DEVELOPMENT OF A NEURAL NETWORK-BASED SPEECH ENHANCEMENT SYSTEM

A Dissertation
Presented to
The Academic Faculty

By

Babafemi O. Odelowo

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

August 2018

DEVELOPMENT OF A NEURAL NETWORK-BASED SPEECH ENHANCEMENT SYSTEM

Approved by:

Dr. David V. Anderson, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Elliot Moore II
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Pamela Bhatti
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Aaron Lanterman
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Brani Vidakovic
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: May 14, 2018

To Tunrola, Ebun, and Tomi.

ACKNOWLEDGEMENTS

I would first like to express my sincere gratitude to my advisor, Dr. David Anderson. He welcomed me into his lab without reservation even though I had not previously worked with him, and I had not been in any of his classes. He also allowed me the freedom to pursue my research interests and was always ready to offer his help and perspective on problems.

I would also like to thank the members of my thesis committee, Dr. Elliot Moore, Dr. Pamela Bhatti, Dr. Aaron Lanterman, and Dr. Brani Vidakovic for their willingness to serve, and for their comments and suggestions which have been instrumental in improving the quality of this dissertation.

I had the privilege of working in the Efficient Signal Processing (ESP) Laboratory with several exceptional people. Dr. Muhammad Rizwan and I had several discussions about machine learning and had a successful collaboration. I will always appreciate Dr. Kaitlin Fair for her brutal honesty and openness in discussions. Several other members of the lab including Dr. Chu Meh Chu, Dr. Jinwoo Kang, Dr. Brad Whitaker, Nathan Parrish, Brandon Carroll, Lee Richert, Chieh-Feng (Jeff) Cheng, You Wang, Chuyao Feng, and Joseph Aribodo provided information on tools and helpful tricks, and gave helpful feedback on some of my presentations. I have fond memories of my time in the ESP lab because of all of their contributions.

I owe a great debt of gratitude to several past and present members of the Electronic Warfare and Analysis Branch (EWAB) in the Electro-Optical Systems Laboratory (EOSL) Unit of Georgia Tech Research Institute (GTRI). First, my branch head, Dr. Micah Coleman, ensured I was always employed as a GRA while I conducted my dissertation research. He was steadfast and tireless in his support and is indeed one of the best bosses I have ever worked for. I worked directly with Dr. Phil West, Dr. Sam Shapero, Dr. Rick Coogle, Kyle Harrigan, and Rick Jones. They all readily shared their expertise on different systems

and contributed to my getting work done efficiently. I am also extremely grateful to Dr Chris Edmonds for his constant encouragement as I worked through the arduous task of finishing my dissertation. Dr Chris Walker, Dr. Josh Wells, Dr Yan Wang, and Yokhanan Beck also contributed positively to my tenure in the lab. Some other members of the larger EOSL family were also a great encouragement. In particular, Jason Zutty and I had several discussions, commiserated, and spurred each other on as we strove to complete our research.

Special thanks to our many friends who stood with my family and offered support and encouragement. I would be remiss if I did not specifically thank Ms. Na Keya Bazemore and Ms. Bileni Teklu who were always available to babysit, feed, and chauffeur our kids. Their help allowed the time to focus on my research and, for that, I am eternally grateful.

I am grateful to my brothers, parents, sisters-in-law, parents-in-law, and my larger extended family for their prayers, support, and constant encouragement. They have been with us in the good times and the not so good times and have never ceased to offer support and pride in all my efforts and progress. It indeed takes a village! Special thanks go to Kemmie for all our late night chats and for all the laughs.

Lastly, I must acknowledge my wife, Tunrola, and children, Ebun and Tomi, for their sacrifice, love, and patience. My wife handled my extended absences with grace and remained steadfast especially during the rocky and uncertain times in the program. Ebun and Tomi were and continue to be an extreme source of joy and happiness. Finally, “now to the King eternal, immortal, invisible, the only God, be honor and glory for ever and ever. Amen.”

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Speech has been indispensable for communication for several millennia. Clean speech is vital in several modern-day applications such as aids for hearing-impaired individuals, speaker identification and automatic speech recognition systems, and as an interface for the control of electronics devices. Background noise, present in all realistic environments, affects the performance of these applications, and it desirable to suppress any noise in speech, i.e., enhance the speech. Although several approaches have been proposed over the past decades, there has been a recent focus on the use of data-driven methods and neural networks, in particular, have shown considerable promise.

Neural networks are powerful machine learning models that have, in the last few years, been applied to several audio and speech signal processing problems including speech enhancement. Although, neural network-based speech enhancement approaches have outperformed traditional model-based approaches, there remain several unanswered questions such as the most suitable network architectures, input features, training targets, and best practices for obtaining optimal results.

This dissertation studies two approaches to the development of a neural network-based speech enhancement system. First, we investigate the use of the extreme learning machine (ELM), an algorithm that allows feed-forward networks to be quickly trained and provides good generalization, for speech enhancement. We explore spectral mapping and time-frequency (T-F) masking approaches and show that the T-F masking approach is the superior ELM-based approach.

The speech enhancement problem is a multivariate regression problem. We show that the solution produced by the ELM is not optimal for multivariate regression as it ignores correlations between the dimensions of the target. We then propose modifications to the extreme learning machine to increase its prediction accuracy on multivariate datasets and demonstrate the improved performance of these algorithms using a synthetic dataset and

several real-world datasets. We also use the improved algorithms in the enhancement of noisy speech and compare the results to those obtained with the original ELM algorithm. Lastly, we compare the performance of ELMs and networks trained conventionally with the back-propagation algorithm.

Neural network-based speech enhancement approaches aim to estimate features of the clean or noise-free speech signal from noisy speech features. These approaches, however, perform poorly in low signal-to-noise ratio (SNR) conditions. With a view to obtaining improved low SNR performance, we develop a noise prediction and time domain subtraction framework for speech enhancement. We extend the development of the noise prediction framework by investigating different training targets and the use of noise-aware training methods and show using objective performance metrics that the proposed framework compares favorably with conventional speech prediction approaches in enhancing speech quality and intelligibility in both seen and unseen noise conditions.

Post-processing techniques can be used to improve the clean speech estimates produced by a neural network. We propose a T-F mask-based post-processing approach for deep neural network (DNN) enhanced speech and show the method always improves both speech quality and intelligibility, and that these improvement are statistically significant. In addition, we analyze the enhanced speech and show that post-processing reduces severe amplification distortions in the magnitude spectrum of the enhanced speech, however, this comes at the cost of slight increase in severe attenuation distortions.

CHAPTER 1
INTRODUCTION

1.1 Motivation

Speech is essential for human communication. Through the use of speech, we express our wants and desires, give instructions, and exchange information. The use of complex speech patterns, including diverse languages, differentiates human beings from other species.

Speech is also increasingly important as a human-machine interface. Automatic speech recognition (ASR) systems like Apple's Siri and Google's Now are commonly used to interface with smartphones. Similarly, smart speakers like Amazon Echo and Google Home, and their ever-ready, nifty virtual assistants, Alexa and Google Assistant respectively, are queried and controlled through spoken dialog. In addition, it is now commonplace to encounter interactive voice response (IVR) systems when placing a phone call to the customer service line of a financial institution or large company.

Speech is seldom produced in an environment devoid of other sources of sound. In an office, spoken dialog may have to compete with the noise of computer keyboards, fans, or other speakers in the background. Similarly, the noise of a construction crew at work, cars passing on a street, or an airplane flying overhead are common disturbances that are encountered in daily life. The noise produced by these sources mixes with speech, degrades it, and, consequently, impairs communication. High noise levels for extended periods can cause a listener to lose concentration and, consequently, miss details of a conversation, a condition known as listener fatigue [1]. Similarly, background noise affects the performance of ASR systems and interferes with the use of smart speaker and similar entertainment systems [2, 3, 4]. It is therefore desirable to have a means of reducing, or even entirely eliminating, the background noise in speech without distorting or compromising

speech, i.e., *enhance* the speech.

Speech enhancement is concerned with improving certain perceptual aspects of speech that has been degraded by additive noise. It has remained an important research problem for several decades due to its importance in applications such as personal and mobile communications, teleconferencing, and the design of aids for individuals with impaired hearing. In addition, speech enhancement plays an important role in robust machine learning applications like speaker identification systems and robust automatic speech recognition systems (ASR) due to the degrading effect of noise on the performance of such systems [5, 6, 7]. Speech enhancement algorithms have traditionally focused on improving speech quality by reducing or suppressing background noise and have thus been referred to as noise suppression algorithms [8, 9]; however, there has recently been increased interest in also improving the intelligibility of noise-corrupted speech [10, 11].

Speech enhancement techniques can be classified as being either single-channel or multichannel [1, 12]. Single-channel or monaural techniques are applicable when the speech signal is acquired using a single microphone, while multichannel techniques are applicable when either two or more microphones, or a microphone array, is used to acquire the speech signal. Single-channel techniques, which must both estimate and suppress noise from a single source, are generally considered more difficult than multichannel techniques which have more degrees of freedom from the spatial diversity provided by the multiple signal sources [12]. Multichannel techniques, have recently seen increased attention due to the availability of more efficient array designs such as superdirective arrays [13], their applicability in devices such as the Amazon Echo, and the release of the Alexa microphone array kit including beamforming and voice processing software technology by Amazon [14]; however, single-channel techniques remain important due to size, weight, and power (SWaP) restrictions in several consumer devices such as cellular devices and hearing aids. This dissertation focuses on single-channel speech enhancement.

Several classical algorithms have been proposed for speech enhancement. These in-

clude spectral subtraction [9], the Wiener filter and its several variants [8, 15], and minimum mean-square error (MMSE) algorithms [16, 17, 18]. While these algorithms have been somewhat successful in suppressing noise, they are typically fraught with artifacts and perform poorly with non-stationary noise [8, 19].

In recent years, there has been increased interest in the use of data-driven methods in several traditional signal and image processing domains such as speech recognition, computer vision, medical image processing, and communication signal processing, and the field of speech enhancement has been no exception. Independent component analysis (ICA) was used to learn a clean speech basis [20], and signal and noise bases were learned using non-negative matrix factorization (NMF) [21, 22]. Speech enhanced using these learned bases was shown to be of superior quality to speech enhanced using a state-of-the-art Wiener filter implementation. Several neural network models including denoising autoencoders [23, 24], deep neural networks (DNNs) [19, 25], and recurrent networks (RNNs) [7, 26] have also been used in speech enhancement frameworks. The neural network models, in particular, have in general shown superior performance in challenging acoustic conditions and can be considered to provide state-of-the-art performance.

Although there has been significant progress, the use of neural networks in speech enhancement is still in nascent stages, and the full potential is yet to be realized. There is no consensus, for instance, on a set of best practices, and several questions such as the best training features, training targets, and network architectures remain. This is in marked contrast with domains such as computer vision and speech recognition. In the computer vision field, for instance, a couple of decades of research into handwriting recognition has resulted in the convolutional neural network (CNN) which exploits an understanding of the human visual cortex to produce good features [27, 28]. The use of these networks with the large datasets and greater computing power available today now produce state-of-the-art results [29]. In a similar manner, the current state-of-the-art speech recognition systems based on DNNs benefit from the decades of research into the Gaussian mixture

model-hidden Markov model (GMM-HMM) systems they supplanted [30].

Neural network-based approaches to speech enhancement have focused almost exclusively on the use of deep neural networks (DNNs). Deep networks are powerful models, however, they are generally difficult to train and require large amounts of data in order to learn good generalizations [30, 31, 32, 33]. The availability of the massive amounts of data required, and the ability to endure long training times is often assumed by researchers. There are, however, several applications in which obtaining large amounts of data is tedious, costly, or simply infeasible. Deep networks, with a larger footprint and higher power consumption, might also be unsuitable for mobile applications or embedded applications when continuous operation or sensing is required as the power requirements of these networks could quickly deplete power sources. It is therefore reasonable to investigate whether simple computing frameworks such as networks with random weights [34] or reservoir computers [35, 36, 37] can be used effectively for a large-scale task such as speech enhancement.

Another area that needs addressing is the performance of DNNs in enhancing low signal-to-noise ratio (SNR) speech signals. The problem arises in part due to the non-stationary nature of the speech signal. The speech signal consists of a series of segments with varying energy. For signals with low-average SNR, low energy segments will tend to be dominated by noise and thus appear noise-like. Consequently, it would be difficult for a DNN to distinguish between the noisy speech and noise segments [38]. In addition, several speech segments, particularly those corresponding to unvoiced speech, are aperiodic and noise-like in nature [8]. Further study is therefore necessary to determine which neural network architectures, speech features, and training targets will be most suitable for enhancing speech in a wide variety of environmental conditions.

1.2 Objectives

The objective of this dissertation is to systematically develop an efficient neural network-based speech enhancement framework. Two different approaches are investigated in pursuit of this objective. In the first approach, we examine the use of extreme learning machines for speech enhancement with the goal of determining the efficacy of neural networks architectures that do not require large datasets and learn quickly. In the second approach, we develop an architecture based on the estimation of the noise or unwanted interference. An overview of the tasks undertaken in pursuit of these objective follows:

- *Design and evaluation of a speech enhancement framework based on the extreme learning machine (ELM)*

  As mentioned in Section 1.1, contemporary neural network-based speech enhancement approaches require large amounts of training data and, consequently, have long training times, often of the order of several days. In addition, these deep learning approaches might not be suitable in all applications. We therefore develop a framework for speech enhancement based on the ELM. We examine the use of spectral mapping and time-frequency (T-F) masking paradigms with ELM. In addition, we study the effect of variables such as data normalization schemes; the size of the network, and the size of the training dataset. The performance of the system is also compared to the performance of a statistical model-based minimum mean-square enhancement algorithm.

- *Develop of a multivariate extreme learning machine*

  We investigate the performance of the ELM with multivariate or vector targets and show the ELM algorithm is not optimal for real-valued, multivariate targets as correlations between the response variables are ignored. We propose two improved ELM algorithms that take into account the correlation between response variables and demonstrate their performance with a variety of datasets.

- *Design of a noise prediction speech enhancement framework*

  Conventional neural network-based speech enhancement systems aim to estimate features of the clean or noise-free speech from noisy signal input features. The use of machine learning techniques, however, affords us the opportunity to estimate other targets, including those of the noise or interference. We investigate the use of a noise prediction-based neural network architecture for speech enhancement. We also develop a novel time-domain noise subtraction speech enhancement framework: a time-domain estimate of the noise signal is synthesized from noise features predicted by the neural network. The enhanced speech signal is consequently obtained by subtraction in the time-domain. We also compare variations of the noise prediction architecture based on time-domain subtraction and spectral subtraction in order to quantify the effect of using the phase of the noisy speech signal for reconstruction of the enhanced speech signal. In addition, we compare the performance this new architecture to that of conventional speech-feature prediction models.

- *Design and evaluation of training targets for the noise prediction framework*

  The careful choice of a training target, the computational goal of a neural network, is an important factor in any supervised learning task. The importance is magnified in speech enhancement because speech enhancement networks are regression models that aim to accurately predict a target function. Although spectral features might seem to be the natural choice for noise prediction, we investigate the use of mask-based features for noise prediction and reconstruction. We introduce a new target, the noise-ratio mask, compare the performance of the different training targets, and determine which targets are most suitable for noise prediction networks. Finally, we compare the performance of the noise targets to that of conventional speech-feature prediction models.

- *Investigate the performance of a speech enhancement framework using multiple neu-*

ral networks

We investigate the performance of the noise prediction architecture which we have developed and refined in the noise-aware training paradigm [6]. We first investigate the use of a fixed or static noise estimate to improve the robustness of the noise prediction models. We then cascade multiple neural networks in an architecture in which one network, the noise estimator, provides the second network, the noise predictor network, with a dynamic estimate of the noise present in the noisy speech to be enhanced. The goal is to develop an approach that would use a neural network for dynamic noise estimation and would also be fully data-driven, i.e., rely on the trained networks without the use of thresholds that could compromise performance.

- *Investigate the performance of a post-processing approach for DNN-based speech enhancement*

  As previously mentioned, speech enhancement networks are regression models that estimate a target function from noisy speech features. Although neural networks have far exceeded the performance of earlier methods such as the minimum mean-square error (MMSE) estimators, the estimates obtained from these networks are still not perfect and could be improved. We seek to develop a simple method that can be used to refine the estimates obtained from a neural network without resorting to further training of the network and without an expansion of either the input or target features. We propose a mask-based processing approach that is easily implemented, compare the performance of a systems with and without post processing, and undertake a rigorous statistical analysis of the results.

1.3 Organization

The rest of this dissertation is organized as follows: in the next chapter we provide background information on select topics germane to the upcoming chapters in this dissertation. In particular, we review approaches to speech enhancement, extreme learning machines,

and multivariate regression techniques.

Chapter 3 is divided into two parts. In the first part, we present a spectral mapping approach to speech enhancement using the ELM, while the second part focuses on a T-F masking approach. We compare both approaches and show that the mask-based approach is superior to the spectral mapping approach.

In chapter 4 we present two proposed extensions of the ELM algorithm that are designed to improve the performance of the ELM on multivariate datasets and compare the performance of these algorithms to the baseline ELM algorithm.

In chapter 5, we evaluate the performance of one of the improved ELM algorithms on the speech enhancement task. We compare the performance of the baseline ELM and improved ELM algorithm, and also compare ELMs and neural networks trained conventionally using the back-propagation algorithm.

In chapter 6, we introduce the speech enhancement framework based on noise prediction. We examine variations of the framework based on time-domain and spectral subtraction and present results comparing the performance of the proposed framework to that of a conventional speech-prediction network.

In chapter 7,we study the performance of three targets in the noise prediction framework introduced, and we show that the mask-based targets are superior to the spectral target in this framework.

In chapter 8, we study the use of noise-aware training strategies as a means of improving the robustness of the noise prediction networks.

In chapter 9, we propose a post processing method for improving the quality and intelligibility of DNN enhanced speech.

Conclusions, including contributions and suggestions for future work are presented in chapter 10.

CHAPTER 2

BACKGROUND

In this chapter, we being by giving an overview of overview of monaural speech enhancement techniques. This is followed by an introduction to the extreme learning machine (ELM). We then conclude by posing the ELM training problem as a multivariate regression problem and review multivariate regression approaches.

2.1 Speech Enhancement

The earliest speech enhancement algorithms were largely intuitive and easily implemented. One of these was spectral subtraction [9]. The method was based on the simple principle that if the corrupting noise is assumed to be additive, then an estimate of the noise-free signal spectrum could be obtained by subtracting an estimate of the noise spectrum from the noisy speech spectrum. Mathematically, the noisy signal, $y(n)$, can be represented as being composed of the clean speech signal, $s(n)$, and the additive noise signal, $d(n)$, as follows:

$$y(n) = s(n) + d(n), \tag{2.1}$$

where n is the sample time. In order to perform the subtraction in the spectral domain, we take the discrete-time Fourier transform of (2.1), and the resulting relationship can be expressed as

$$Y(\omega) = X(\omega) + D(\omega) \tag{2.2}$$

where $Y(\omega)$, $X(\omega)$, and $D(\omega)$ are the complex spectral representations of the noisy speech, clean speech, and additive noise respectively. If we employ polar representation of the

quantities in (2.2), the spectral subtraction process can be summarized as

$$\begin{aligned} \hat{X}(\omega) &= |Y(\omega)|e^{j\phi_Y(\omega)} - |\hat{D}(\omega)|e^{j\phi_D(\omega)} \\ &\approx (|Y(\omega)| - |D(\omega)|)e^{j\phi_Y(\omega)}, \end{aligned} \tag{2.3}$$

where $|\hat{D}(\omega)|$ is an estimate of the additive noise magnitude spectrum which is obtained by computing the average value of the noisy speech spectrum in regions where speech activity is not present. With spectral subtraction, there is the potential of obtaining negative magnitude spectral values due to inaccuracies in the noise estimation process, however, these problems are typically solved using half-wave rectification of the estimated clean speech spectrum. The major drawback of spectral subtraction was a residual noise artifact termed "musical noise" [39]. Improvements to the basic spectral subtraction process employed heuristics to combat musical noise [39]. These changes were able to reduce, but not completely eliminate musical noise. Furthermore, the use of over-subtraction of the estimated noise in this algorithm typically resulted in distortion of the lower energy portions of the speech signal [8]. Alternate techniques including nonlinear, multiband, and geometric approaches have been proposed to deal with some of the deficiencies of spectral subtraction [8, 40, 41, 42]

The next generation of speech enhancement algorithms was based on mathematical models of the speech and corrupting noise signals and derived optimal estimates of desired speech parameters. These include the Wiener filter [43], which was first applied to speech enhancement by Lim and Oppenheim [15, 44], and the minimum mean-square error (MMSE) short-time spectral amplitude [16] and MMSE log-spectral amplitude (LSA) [17] estimators commonly referred as the Ephraim-Malah models. Although the use of these methods resulted in lower distortion and little to no musical noise under some noise conditions [45], they typically failed to improve speech quality in non-stationary environments that typified real-world conditions.

Researchers have proposed variants of the Wiener filter including the iterative [44], constrained iterative [46], and codebook driven Wiener filters [47]. Similarly, MMSE models that use statistical distributions that more accurately reflect empirical speech data have been proposed. These include models based on the Gamma [48], supergaussian [49], and Laplacian distributions [50]. Improvements have also been made to the MMSE models by incorporating speech-presence uncertainty or probability of speech absence. These include the McAulay and Malpass soft decision estimator [51], the multiplicatively-modified LSA estimator [52], and the optimally-modified LSA estimator by Cohen [18, 53]. The latter estimator is considered a state-of-the-art statistical model and is often used as a performance benchmark [19].

Another class of algorithms consists of those based on psychoacoustic principles. These algorithms aimed to eliminate only the audible portion of the noise instead all of the additive noise. The rationale behind these algorithms was that imperceptible noise did not affect the perceptual characteristics of the speech and thus did not need to be removed. A seminal perceptual model was the modification of spectral subtraction by Virag [54]. In this algorithm, the calculation of the of the over-subtraction parameters used in [9] were modified to incorporate a noise masking threshold based on the principle of simultaneous masking [55]. Other perceptual models in literature include models exploiting temporal masking [56, 57], the audible noise algorithms [58, 59], and the perceptual Wiener filter models described in [60, 61]. These models were, in general, more successful than the models they derived from in eliminating musical noise and thus providing a more pleasurable listening experience [8].

The need for mathematically tractable speech and noise models often restricted the all of the previously described algorithms to rely on unfulfilled assumptions [8, 62]. Their performance was also poor in non-stationary noise. Neural networks are attractive models for the speech enhancement task as they are not limited by the need for tractable mathematical models to describe the interaction between the speech and corrupting noise signals. The

use of neural networks in speech enhancement problem is not a new undertaking. Early works include those by S.Tamara [63, 64], Xie and Compernolle [65], and Sorenson [66]. These early networks were small and used very small datasets, nonetheless, both sets of authors reported successful noise suppression with these networks.

In recent years, several researchers have tackled the speech enhancement problem with neural networks. Deep networks, in particular, have been a common choice due to the successes of these models in several speech and computer vision problems [29, 30]. Network architectures including deep denoising autoencoders [23, 24], deep neural networks [19, 25, 67], and recurrent networks have been used [7, 26]. Neural networks have also been used in conjunction with some of the classical methods such as in [23], where the authors incorporated a weighted denoising autoencoder as part of a signal-to-noise (SNR) estimation module leading to a Wiener filter.

DNN-based speech enhancement models are regression models that learn a mapping between noisy speech input features and a desired target. The training process is a supervised learning process [68]. Three broad paradigms have emerged in the literature, namely, spectral mapping [19, 67], time-frequency (T-F) masking [7, 25, 69], and multitask learning approaches [38, 70]. In the spectral mapping approach, noisy and noise-free spectral input pairs are used as training inputs and targets (responses) respectively. During the enhancement or evaluation phase, the neural network predicts noise-free spectral estimates from noisy input spectra, and the noise-free speech is reconstructed from the predicted spectral estimates. T-F masking approaches such as the ideal binary mask (IBM) and ideal ratio mask (IRM) use a trained neural network to estimate a T-F weighting function from noisy input features. The masking function is applied to the noisy speech spectra and noise-free speech is reconstruction from the filtered spectra. Multitask learning approaches, on the other hand, use a trained neural network to jointly estimate clean log power spectra and secondary features such as mel-frequency cepstral coefficients (MFCCs), binary mask targets, and SNR. In an extensive study of training targets for speech separation [69], two

masking targets, the ideal ratio mask (IRM) and the short-time Fourier transform mask (FFT-MASK), were shown to be superior to other training targets. The same conclusion about the IRM was reached in a later study on enhancement of noisy and reverberant speech [25]. In another study [19], however, spectral training targets were shown to be superior to both IRM and FFT-MASK targets. This is still an open issue in the literature.

While the neural network based architectures have vastly outperformed the classical algorithms [19], low SNR performance remains an open problem. One approach taken by the authors in [38] is to use multiple networks including a voice activity detector network which essentially provides speech/non-speech probabilities for each frame of speech. Although the authors report improvement over the baseline model that uses a single neural network, further work is still needed in this area. In this dissertation, we design and evaluate an approach geared towards improving the low-SNR performance of speech enhancement system.

2.2 Extreme Learning Machines

Extreme learning machines are feed-forward neural networks in which the synaptic weights are learned without the use of iterative tuning methods [71, 72]. ELMs are similar to two earlier architectures: networks with random weights by Schmidt *et al.* [34], and random vector functional link (RVFL) networks by Pao *et al.* [73]. They could be shallow or deep networks, and can be applied to both regression and classification problems [74, 75].

ELMs are trained by assigning input weights and biases of the first hidden layer randomly generated values. Weights of subsequent layers are then obtained by direct, closed-form least-squares optimization. The use of direct computation in the learning of weights gives two advantages, namely, short training times and good generalization capability with smaller training datasets [71, 74].

Consider a single-hidden layer network with L hidden nodes and a training dataset consisting of N distinct input-output pairs $(\mathbf{x}_i, \mathbf{t}_i), i = 1, \ldots, N$ where $\mathbf{x}_i, \in \mathbf{R}^n, \mathbf{t}_i, \in \mathbf{R}^m$

are respectively the input data and target output vectors. The output of the network for any of the input vectors can be represented as

$$f(\mathbf{x}_j) = \sum_{i=1}^{L} \boldsymbol{\beta}_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i), \qquad j = 1, \ldots, N \tag{2.4}$$

where $\mathbf{w}_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$ is the vector of weights between the input nodes and the i^{th} hidden layer node, $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ is the vector of weights between the i^{th} hidden layer node and the output nodes, b_i is the i^{th} hidden layer node bias, $\mathbf{w}_i \cdot \mathbf{x}_j$ is the inner product between the vectors $\mathbf{w}_i$ and $\mathbf{x}_j$, and $g(x)$ is an activation function.

The weights $\mathbf{w}_i, i = 1, \ldots, L$ and biases, $b_i, i = 1, \ldots, L$ are randomly assigned, and the ELM algorithm seeks to find output weights $\boldsymbol{\beta}_i, i = 1, \ldots, L$ such that

$$\sum_{i=1}^{L} \boldsymbol{\beta}_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{t}_j, \qquad j = 1, \ldots, N \tag{2.5}$$

These equations can be represented in matrix notation as [71]:

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{2.6}$$

where

$$\begin{aligned} &\mathbf{H}(\mathbf{w}_1, \ldots, \mathbf{w}_L, b_1, \ldots, b_L, \mathbf{x}_1, \ldots, \mathbf{x}_N) \\ &= \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \ldots & g(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \ldots & g(\mathbf{w}_L \cdot \mathbf{x}_N + b_L) \end{bmatrix} \end{aligned} \tag{2.7}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_L \end{bmatrix}, \qquad \text{and} \qquad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}. \tag{2.8}$$

The matrix, $\mathbf{H}$, termed the "hidden node matrix" [71], is thus the output of the hidden nodes for all the training inputs, and $\mathbf{T}$ is the matrix of training targets.

Given an assigned set of hidden weights and biases, the network is trained by finding weights, $\hat{\boldsymbol{\beta}}$, such that,

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{H}(\mathbf{w}_1, \ldots, \mathbf{w}_L, b_1, \ldots, b_L)\boldsymbol{\beta} - \mathbf{T}|| \tag{2.9}$$

The output weights are thus obtained as

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T} \tag{2.10}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of the matrix $\mathbf{H}$.

An alternate form of the solution is obtained if a regularization parameter, λ, is included to prevent overfitting. ELM learning theory aims to obtain the minimum training errors well as the smallest norm of the output weights [74]. This can thus be expressed as [75]

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \quad \{||\boldsymbol{\beta}|| + \lambda||\mathbf{H}(\mathbf{w}_1, \ldots, \mathbf{w}_L, b_1, \ldots, b_L)\boldsymbol{\beta} - \mathbf{T}||\}\,, \tag{2.11}$$

and the solution is obtained as

$$\hat{\boldsymbol{\beta}} = \left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{T}. \tag{2.12}$$

2.2.1 The Online Sequential ELM

The ELM formulation as presented in (2.6) - (2.10) is a batch learning algorithm. As such, the ELM can only be used in problems in which the entire training dataset can fit into computer memory. An alternate formulation, the online sequential ELM (OS-ELM) [76], which results from the application of recursive least-squares algorithm to (2.6), addresses this shortcoming. The OS-ELM consist of two phases: an initialization phase in which

a portion of the data is chosen for processing based on the number of hidden nodes in the network, and a sequential phase in which the remaining training data can be processed either in single instances, or in fixed or variable size batches. An alternate online sequential algorithm based on orthogonal least squares is presented in [77]. This algorithm has the disadvantage that it can only process single data instances and thus could be slow on larger datasets.

2.3 Multivariate Regression

The ELM training problem (2.6) is a multivariate regression problem. Multivariate regression is alternately known as multi-target, multi-output, or multi-response regression in the statistical literature [78]. The problem can be summarized as follows: given a training data set D of N instances, $D = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(N)}, \mathbf{y}^{(N)})\}$, where the predictors $\mathbf{x}^{(l)} \in \mathbf{R}^m, l \in \{1, \ldots, N\}$ and corresponding target values $\mathbf{y}^{(l)} \in \mathbf{R}^d, l \in \{1, \ldots, N\}$ are m- and d-dimensional vectors respectively, find a function $\mathbf{h} : \mathbf{x} \rightarrow \mathbf{y}$ that assigns each instance $\mathbf{x}$ to its corresponding target value $\mathbf{y}$. Borchani et al. [78] categorize multivariate regression methods as either problem transformation or algorithm adaption methods. Problem transformation methods decompose the problem into d single-target or uni-variate regression problems. These methods consequently ignore any correlations between the response variables and are not optimal. Algorithm adaptation methods, on the hand, aim to predict all the targets or response variables using a single model. This enables the model to capture all dependencies between the output variables and thus ensures better predictive performance when targets are correlated [78, 79].

A few algorithm adaptation methods have been proposed in the literature. Brown and Zidek [80] proposed adaptive multivariate ridge regression. This method adapts the familiar uni-variate ridge regression [81] result to the multivariate problem. It however produces an undesirable expansion of the problem and could only be applied in problems where the dimension of the response, d, and the number of predictors, m, is small. Other methods in

the literature are the reduced-rank regression [82] and filtered canonical y-variate regression (FICYREG) [83]. Breiman and Friedman proposed the curds & whey method [79]. There are two variants of the method. In each variant, d single target responses are computed and then cross-validatory shrinkage based on canonical correlations [84] between the variables is applied. They showed these methods outperformed the method of performing d single target regressions (that ignored correlation between variables), and several approaches proposed earlier in the literature including the ones mentioned above. The curds & whey method has also been extended to non-linear regression by D'Ambra and Lombardo [85].

Another algorithm that has been adapted for multivariate regression is support vector regression (SVR) [86]. SVR is an extension of the support vector machine framework [87], which was originally developed for binary classification problems [88], to regression and functional estimation problems [89, 90]. Two versions of this adaptation can be found in the literature. In the first named multi-dimensional SVR (MSVR) by Perez-Cruz et al. [91], the authors use an extension of Vapnik's ϵ-insensitive loss function [90] in a multivariate context. A second version, named multiregressor SVR (M-SVR) by Sanchez-Fernandez et al. [92], or multi-output SVR (M-SVR) by Tuia et al. [93], replaces the ϵ-insensitive loss function with an L_2-based norm. The L_2-based norm in the latter version of the algorithm has the advantage that it is differentiable. The algorithm was used for nonlinear channel estimation in multiple-input multiple-output (MIMO) systems [92], and for parameter estimation in a remote sensing application. [93].

CHAPTER 3

SPEECH ENHANCEMENT USING EXTREME LEARNING MACHINES

This chapter examines the use of the extreme learning machine for speech enhancement. The work presented in this chapter has been published in the *Proceedings of the 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* [94] and in the *Proceedings of the 2017 Asilomar Conference on Signals, Systems, and Computers* [95].

3.1 Introduction

In this chapter we develop a framework for speech enhancement using the extreme learning machine (ELM). Various neural network architectures, as previously mentioned in Section 2.1, have been successfully applied to the speech enhancement problem. These networks, including denoising autoencoders [23, 24], deep neural networks [19, 67, 25], and recurrent networks [7, 26], have in general shown impressive performance in a variety of challenging noise conditions.

While the previously mentioned network architectures have performed well in speech enhancement, these performances come at a huge cost. Firstly, large training datasets are required. These could be up to several hundred hours in length [19]. Secondly, extremely long training times are needed due to the use of these large datasets. An alternative neural network architecture, the ELM, has attracted the attention of researchers in several disciplines over the last decade [96]. ELMs are attractive for the speech enhancement task as they can be trained quickly and provide good generalization capability with small amounts of training data [71].

The rest of the chapter is organized as follows: we investigate a spectral mapping approach to speech enhancement using the ELM in Section 3.2. We study the use of different input normalization schemes and the use of regularization in network training. We

also study the effect of network size, the use of context windows, the performance of this framework in matched and mismatched noise conditions, and compare its performance to that of the optimally modified log-spectral amplitude (OM-LSA) algorithm. This work [94] is to be best of the authors' knowledge, the first use of the ELM directly for speech enhancement. A time-frequency (T-F) masking approach to speech enhancement using ELMs is investigated in Section 3.3. We once again study how different parameters affect the efficacy of the approach, and compare the performance of the spectral mapping and T-F masking frameworks. Concluding remarks are made is Section 3.4.

3.2 Spectral Mapping

3.2.1 System Overview

The ELM was a single hidden layer, or shallow, network. The number of nodes in the hidden layer ranged from 2000 - 8000 nodes. The ELM is trained for speech enhancement in a supervised manner using input features computed from noisy speech and corresponding targets computed from noise-free speech. Each training pair, $(\mathbf{x}_i, \mathbf{t}_i)$, is respectively comprised of a corresponding noisy and noise-free log magnitude spectral pair. Both the batch ELM and OS-ELM algorithms were used in training the networks. The choice of algorithm was driven by the memory requirements of the algorithms which, in turn, depended on the number of nodes in the hidden layer. The batch algorithm was used for networks with up to 6000 hidden nodes, and the OS-ELM algorithm was otherwise used.

3.2.2 Experiments

All experiments were performed using recorded sentences from the IEEE Corpus included with the NOIZEUS database [97, 98]. The corpus is comprised of 72 lists, each of which contains 10 sentences. Our noise samples came from a database of 100 non-speech sounds [99]. Both the noise-free speech and noise recordings were resampled to 8kHz. The training dataset was comprised of an appropriate number of sentences from lists 1 - 57 for the

Table 3.1: Description of noise types.

Noise	Description	Noise	Description
n1	Crowd	n6	Water
n2	Machine	n7	Wind
n3	Alarm/Siren	n8	Bell
n4	Traffic/Car	n9	Cough
n5	Animal	n10	Clap

desired training set size, while testing was done with the 50 sentences from lists 68 - 72. Noisy utterances were created by adding 10 types of noise to each of the sentences at six noise levels ranging from 20dB to -5dB in 5dB steps. The noise types are listed in Table 3.1.

Short-time Fourier analysis was done using a Hamming window, 32ms frames, with 50% overlap. We investigated the use of both zero mean, unit variance (ZMUV) and different min-max normalization schemes, and the use of regularization in the training the network. Only the noisy input features were normalized; the noise-free targets were not normalized. To allow the network to take advantage of temporal information, we employed a context window that included features of adjacent speech frames. Consequently, each input vector was constructed as

$$\mathbf{x}_i = [\mathbf{y}_{i-l}, \ldots, \mathbf{y}_i, \ldots, \mathbf{y}_{i+l}] \tag{3.1}$$

where l is the size of the context window, and $\mathbf{y}_i$ is the log spectral feature vector of the i^{th} frame.

Testing involved using the network to predict noise-free spectral estimates from noisy magnitude spectral inputs. The noise-free estimates were combined with the noisy signal phase and the resulting speech signal synthesized using the overlap-add method [100]. The performance of the ELM speech enhancement system was compared to that of the optimally modified log-spectral amplitude (OM-LSA) estimator [18, 62, 53]. For matched

noise testing, the same noise sounds were used for both the training and testing datasets. Mismatched testing, on the other hand, was done using noise types from the same categories given in Table 3.1 that were not included in the training data. Training and testing were performed using all of the 10 selected noise types in both cases. The choice of the ELM training algorithm, as previously mentioned, was based on the size of the hidden layer. The size of the training set was also a determining factor: all training sets of 5 hours or more hours in length were were paired with the OS-ELM.

Results were subjectively evaluated using informal listening tests and objectively evaluated using perceptual evaluation of speech quality (PESQ), a standard perceptual quality measure that has been shown have high correlation with subjective test scores [97, 101]. PESQ scores range between -0.5 to 4.5, with higher scores corresponding to higher perceptual speech quality.

3.2.3 Results and Discussion

Normalization of Input Features

The root mean-square error (RMSE) of the ELM with ZMUV and two min-max normalization schemes is shown in Figure 3.1. The best results were obtained with features normalized to the range [-1,1]. This is because the ELM algorithm computes output weights using a least-squares solution. As such, all the input features should be scaled similarly. It should be noted that while ZMUV normalization is commonly used with gradient-descent based approaches, it is clearly not the best choice for the ELM. The figure also shows that the networks with more hidden nodes are more effective in fitting the data and therefore have lower RMSE.

Use of Weight Regularization

The RMSE of the ELM for different values of the regularization parameter is shown in Figure 3.2. While the use of regularization was shown to be beneficial in some classification

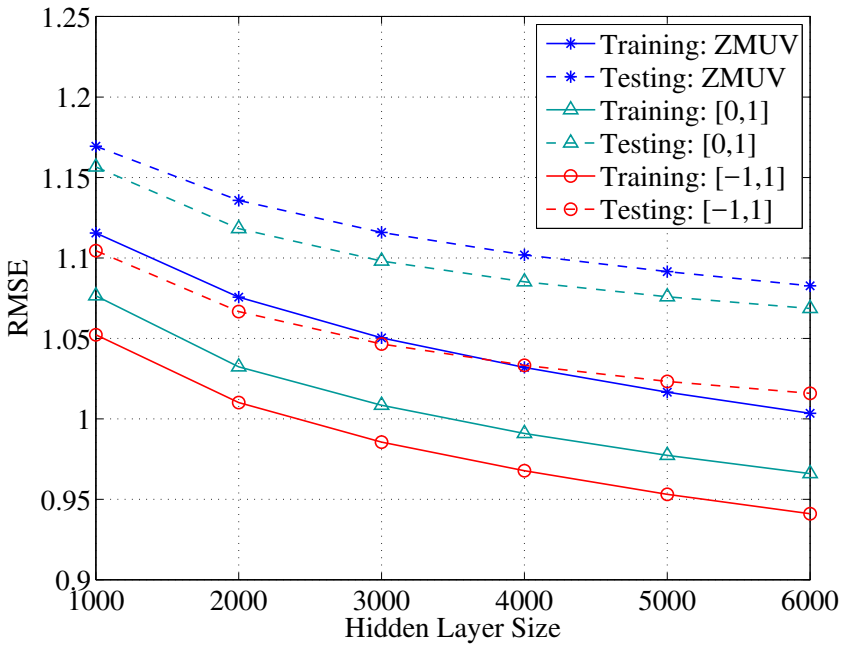


Figure 3.1: Training and testing error for different normalization schemes on a 6000 hidden node network with 2.5 hours of training data.

experiments [74], this was not the case in our study. The performance of the ELM was seen to be stable over a wide range of values of the regularization parameter outside of which performance rapidly declined. Similar stability was noticed with ZMUV normalization, but there was instability when regularization was used with [0,1] normalization.

Effect of Network Size

Table 3.2: Average PESQ scores for networks with hidden layer of different sizes and 10 hrs of training data.

SNR	Matched				Mismatched			
(dB)	Noisy	2000	4000	6000	Noisy	2000	4000	6000
20	3.03	3.28	3.40	3.46	3.18	3.18	3.25	3.29
15	2.70	3.10	3.22	3.29	2.87	2.98	3.03	3.07
10	2.38	2.88	3.00	3.07	2.57	2.75	2.79	2.82
5	2.07	2.62	2.74	2.81	2.29	2.49	2.53	2.55
0	1.79	2.33	2.44	2.52	2.04	2.22	2.27	2.28
-5	1.50	2.02	2.12	2.19	1.78	1.96	2.00	2.01

Average PESQ scores for networks with between 2000 - 6000 hidden nodes are pre-

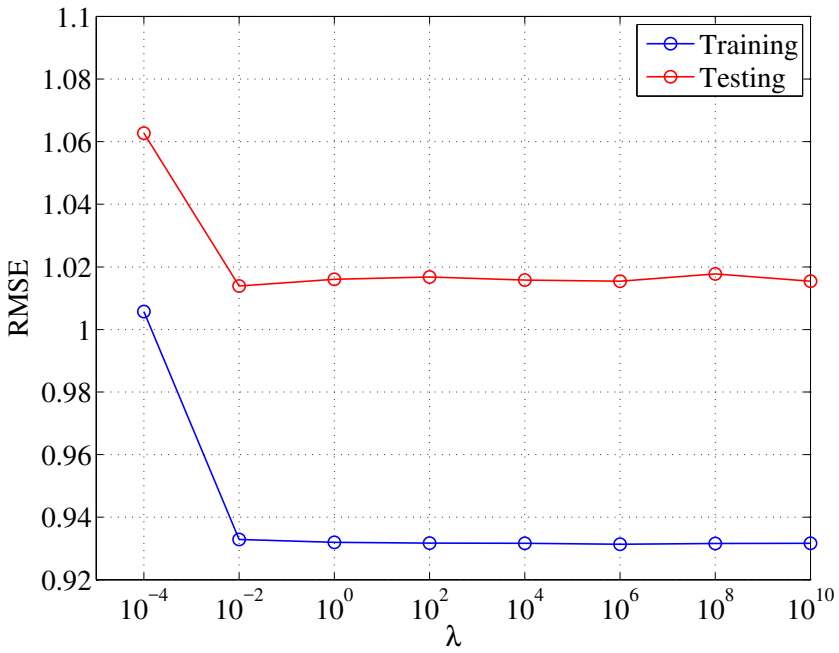


Figure 3.2: Training and testing error with regularization for a 6000 hidden node network with 2.5 hours of training data.

sented in Table 3.2. It can be seen that for both matched and mismatched noise, the larger networks are more effective at noise suppression than the smaller networks. This result mirrors the observations from Figure 3.1 where the larger networks were seen to be more effective at fitting the data. While larger networks were, in general, more effective at noise suppression, the performance began to saturate at 6000 nodes, and there was little benefit from using more hidden nodes. It can be recalled from Section 2.2 that the inputs weights in the ELM are kept fixed. Since only the output weights are trained, the larger networks have more degrees of freedom to learn the relationships between the noisy inputs and noise-free targets and perform better.

Effect of Training Set Size

Average PESQ scores for the ELM trained with datasets of different sizes are presented in Table 3.3. Unsurprisingly, the performance is better with more training data. It is notable, however, that the network is able to improve speech quality with just 1 hour of training data

Table 3.3: Average PESQ scores for a 7000 hidden node network with training datasets of different sizes in matched testing.

SNR	Noisy	1 hr	2.5 hrs	5 hrs	10 hrs	25 hrs
20	3.03	2.94	3.18	3.34	3.42	3.48
15	2.70	2.83	3.03	3.17	3.24	3.29
10	2.38	2.67	2.83	2.95	3.02	3.06
5	2.07	2.47	2.59	2.69	2.75	2.79
0	1.79	2.22	2.31	2.40	2.44	2.48
-5	1.50	1.94	1.99	2.06	2.11	2.15

at all but the highest SNR. Average PESQ scores increase rapidly as the training dataset is increased to 10 hours, but the rate of increase declines as the size is further increased to 25 hours. This suggests that the network generalizes well with the smaller training sets.

Table 3.4: Average PESQ scores for a 7000 hidden node network with training datasets of different sizes in mismatched testing.

SNR	Noisy	1 hr	2.5 hrs	5 hrs	10 hrs	25 hrs
20	3.18	2.83	3.05	3.16	3.22	3.26
15	2.87	2.68	2.86	2.95	2.99	3.02
10	2.57	2.48	2.63	2.70	2.73	2.76
5	2.29	2.24	2.35	2.42	2.44	2.46
0	2.04	1.95	2.04	2.11	2.13	2.14
-5	1.78	1.67	1.74	1.80	1.82	1.84

The average PESQ scores in mismatched noise tests are presented in Table 3.4. Once again, there is a rapid increase in the scores with the increase in size of the smaller datasets, however, the rate of change decreases with the larger datasets. This is the same trend that was observed with the matched noise tests. It can also be noticed that more training data is need to obtain a consistent improvement in speech quality when the training and test noise types are mismatched: about 5 hours of data are needed as opposed to just 1 hour as observed with the matched noise tests.

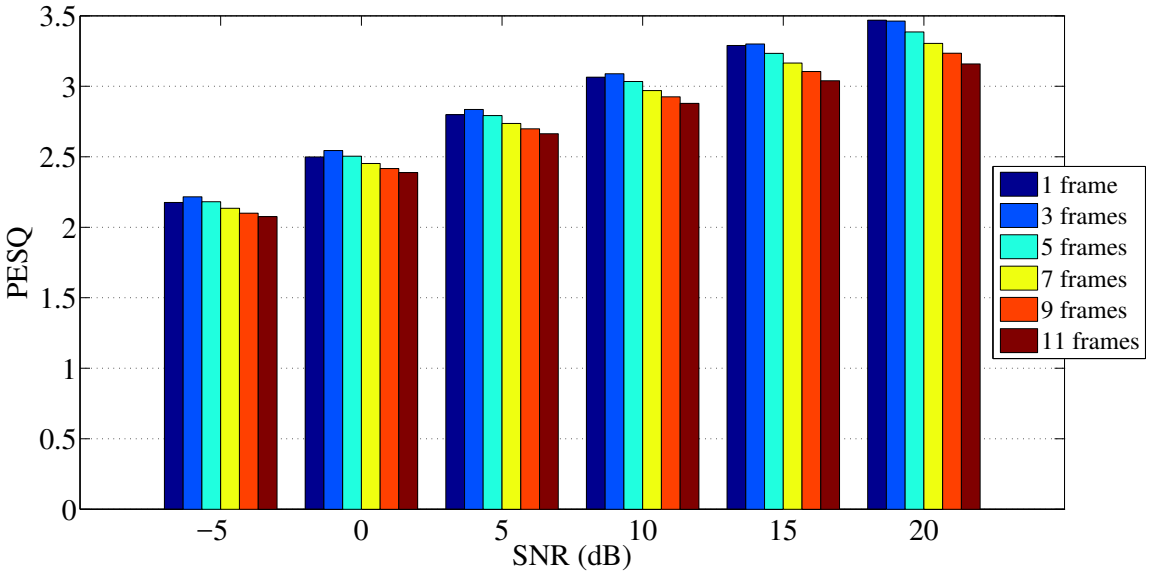


Figure 3.3: Average PESQ scores by SNR for different numbers of context frames.

Effect of Context Length

The effect of employing context on performance in shown in Figure 3.3. While the addition of a single context frame is beneficial, addition of more context frames degrades the performance. This is in contrast with published results for deep networks [19]. The likely reason is that with input weights being fixed, the number of degrees of freedom available to learn information from the larger input vector is unchanged, hence increasing the size of the input vector effectively reduces the size of the network and the performance degrades.

Algorithm Comparison: Matched Noise

The performance of the OM-LSA algorithm and the ELM with matched noise is shown in Figure 3.4. The ELM outperforms the OM-LSA algorithm at all SNR levels with as little as 5 hours of training data. Some further insight can be gained from Figure 3.5 where the results are compared by noise type. OM-LSA is most competitive with the ELM in both crowd and wind noise. Both of these noise types have constant, almost white characteristics which favor the OM-LSA algorithm. The advantage of the ELM with non-stationary noise can, however, be seen with all the other noise types.

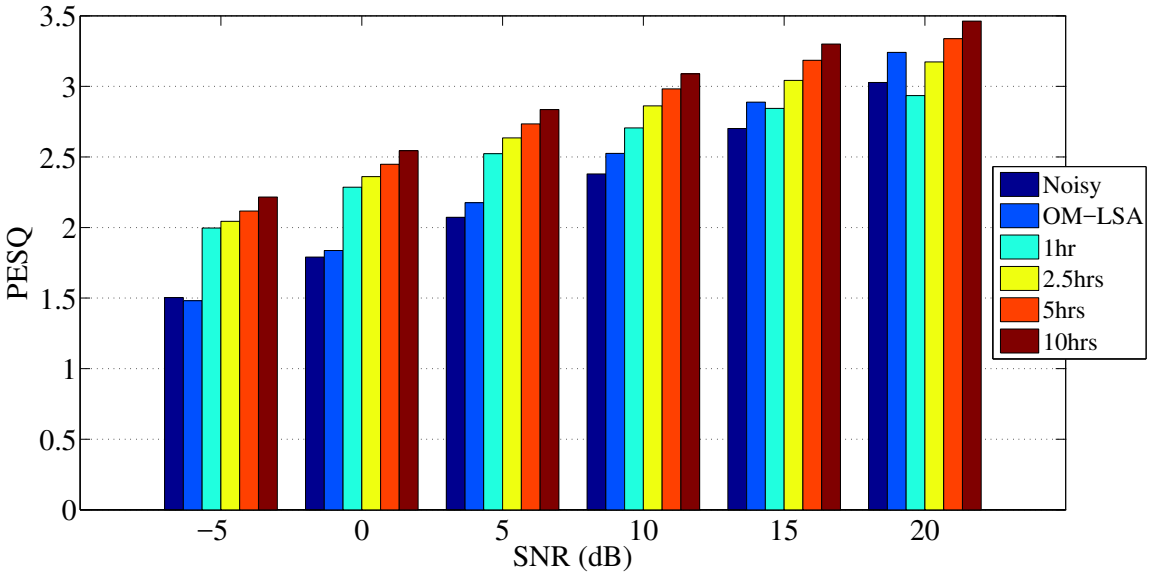


Figure 3.4: Average PESQ scores by SNR for OM-LSA and the ELM with matched training datasets of different sizes.

Algorithm Comparison: Mismatched Noise

The performance of the OM-LSA algorithm and the ELM with mismatched noise is shown in Figure 3.6. OM-LSA performs slightly better at 20 dB SNR, but is worse at all others SNR levels. This underscores the fact that the ELM is able to learn a mapping between noisy and noise-free input-output spectral pairs. It also suggests that the network will be more effective in suppressing unseen noise types if the number of noise types in the training set is increased. We observed that OM-LSA was most competitive against the ELM in the Traffic/Car noise category. The noise used here was also constant and pink. OM-LSA thus has an added advantage as the noise is fairly stationary, and ELM has not learned a mapping for the testing noise type.

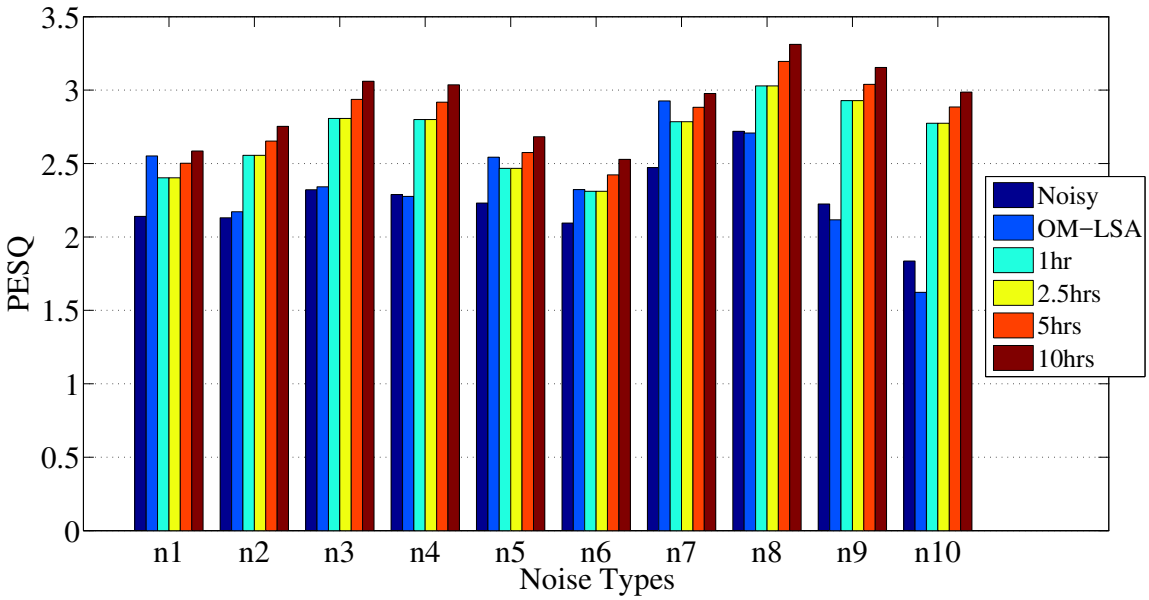


Figure 3.5: Average PESQ scores by noise type for OM-LSA and the ELM with matched training datasets of different sizes.

3.3 T-F Masking

3.3.1 System Overview

The ELM, once again, was a single hidden layer network, and the number of nodes in the hidden layer ranged from 2000 - 8000 nodes. In the T-F masking approach, each training pair, $(\mathbf{x}_i, \mathbf{t}_i)$, is respectively comprised of a log magnitude spectral vector and a time-frequency (T-F) target mask. The ideal ratio mask (IRM) which is defined as [69]

$$IRM(t,\omega) = \left(\frac{S^2(t,\omega)}{S^2(t,\omega) + N^2(t,\omega)} \right)^{\frac{1}{2}} \tag{3.2}$$

where $N^2(t,\omega)$ and $S^2(t,\omega)$ represent the added-noise and speech signal power spectral densities respectively, was the training target.

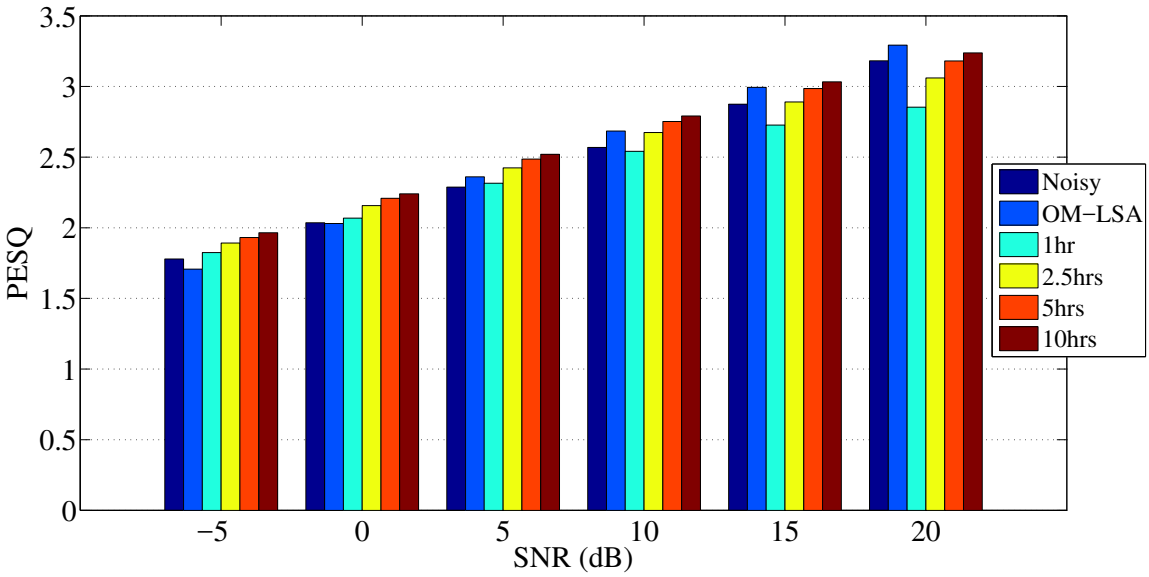


Figure 3.6: Average PESQ scores by SNR for OM-LSA and the ELM with mismatched training datasets of different sizes.

3.3.2 Experiments

We repeated the experiments that were performed with the spectral mapping ELM in the previous section also noise-free recording from the IEEE Corpus/NOIZEUS database and noise samples from the database of 100 non-speech sounds. In addition, we examined the limiting performance of the network with large hidden layer sizes and performed experiments to delve into the reasons why the use of additional context frames worsened, rather than improving, performance as might be expected. The processing of the speech signals was as follows: The speech signals were divided into 32ms frames with 50% overlap, and spectral features extracted from the clean speech, noisy speech, and from the added noise signals were used to create input-output pairs for training the network. Following previous results on normalization of input features, we normalized input features to the range of [-1,1]. The training target, which has a range of [0,1], was not normalized. The construction of the input vectors using context windows was as described previously in (3.1). The performance of the system was evaluated with matched and mismatched noise types using the same noise categories in Table 3.1, and results were objectively evaluated using the PESQ

metric.

3.3.3 Results and Discussion

Effect of Network Size

The root mean-square error (RMSE) of the of the ELM on a training and validation set with matched noise types is shown in Figure 3.7. The training and test RMSE both decrease monotonically as the hidden layer size is increased showing that a larger network is a more effective learner than a smaller network. As stated in Section 2.2, the inputs weights in the ELM are kept fixed during training. Since only the output weights are trained, the larger networks have more degrees of freedom to learn the relationship between the input and target features and consequently perform better. While the training RMSE reduces at an almost consistent rate from about 4000 - 10000 hidden nodes, the rate of reduction in the testing RMSE can be seen to level off at 8000 nodes. This suggests there might not be much of a benefit from using a network with more than 8000 hidden nodes. The PESQ results in Table 3.5 confirm this intuition. The PESQ scores at every SNR level can be seen to increase as the network size is increased, however, the rate of increase in scores becomes smaller as the network size becomes larger.

Table 3.5: Average PESQ scores for networks of different sizes with matched noise types and 10 hours of training data.

SNR (dB)	Noisy	Hidden Layer Size (Nodes)			
		2000	4000	6000	8000
20	3.03	3.45	3.53	3.56	3.58
15	2.70	3.21	3.28	3.32	3.34
10	2.38	2.93	3.00	3.05	3.08
5	2.07	2.63	2.71	2.77	2.80
0	1.79	2.31	2.40	2.46	2.50
-5	1.50	1.98	2.07	2.12	2.17
AVE.	2.25	2.76	2.83	2.88	2.91

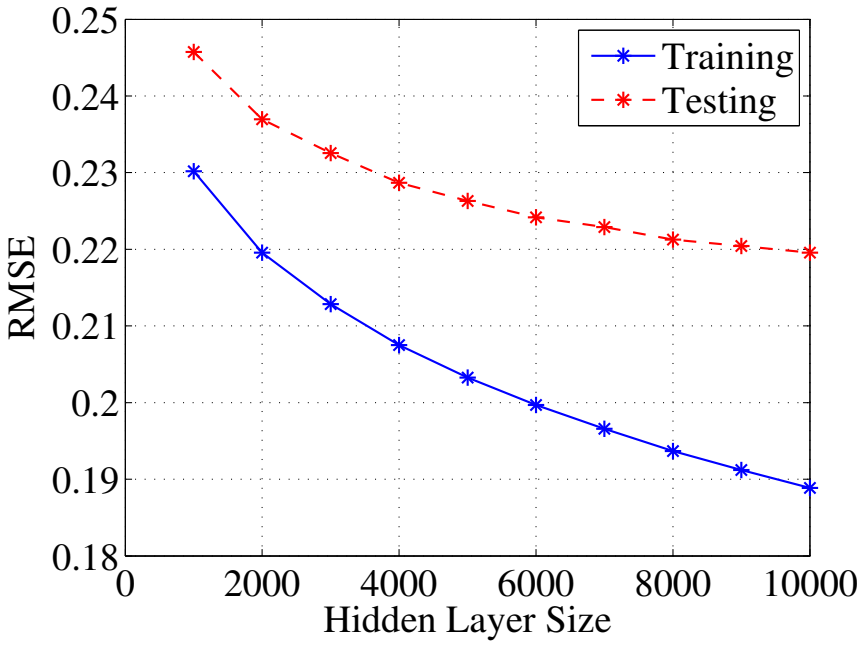


Figure 3.7: Training and testing error using the OS-ELM algorithm on different-sized networks with 2.5 hours of training data.

Effect of Training Set Size

The average PESQ scores for the ELM when trained with datasets of different sizes are presented in Table 3.6. Average scores increase as the size of the training dataset is increased showing that the ELM performs better with more training data. The closeness of the average scores, however, shows that the ELM is particularly effective at suppressing noise when trained with a small dataset, and the network generalizes well with smaller datasets. The results for mismatched noise tests are presented in Table 3.7. The average scores increase as the size of the training set is increased from 1 to 2.5 hours, but there is no further increase in scores as the size of the training set is increased to 10 hours. The difference in average scores can be seen to be even closer than those for matched noise. A training set size of 2.5 hours gives the best results in mismatched noise, hence, we can once again come to the same conclusion that the network generalizes well with smaller datasets.

Table 3.6: Average PESQ scores for a 7000 hidden node network with matched training datasets of different sizes.

SNR (dB)	Noisy	1 hr	2.5 hrs	5 hrs	10 hrs
20	3.03	3.53	3.56	3.56	3.57
15	2.70	3.30	3.33	3.34	3.35
10	2.38	3.05	3.08	3.09	3.11
5	2.07	2.78	2.80	2.82	2.84
0	1.79	2.47	2.49	2.52	2.54
-5	1.50	2.14	2.15	2.18	2.20
AVE.	2.25	2.88	2.90	2.92	2.93

Table 3.7: Average PESQ scores for a 7000 hidden node network with mismatched training datasets of different sizes

SNR (dB)	Noisy	1 hr	2.5 hrs	5 hrs	10 hrs
20	3.18	3.37	3.38	3.38	3.38
15	2.87	3.09	3.11	3.10	3.10
10	2.57	2.80	2.82	2.81	2.81
5	2.29	2.51	2.54	2.52	2.52
0	2.04	2.24	2.25	2.24	2.25
-5	1.78	1.95	1.97	1.95	1.96
AVE.	2.45	2.66	2.68	2.67	2.67

Effect of Context Length

The average PESQ scores when the size of the context window is varied are shown in Figure 3.8. While the addition of one or two frames is beneficial, the addition of more frames degrades the performance. Furthermore, the use of a single context frame on each side of the target frame can be seen to be optimal. This is in contrast with results obtained with deep networks [19]. The results in Figure 3.8 were obtained by increasing the size of the context window while keeping the size of the hidden layer fixed. In order to gain a better perspective of these somewhat puzzling results, we also examined the RMSE of the ELM as the context window and hidden layer size are both increased. The results are shown in Figure 3.9. The training and test RMSE both decrease monotonically as

previously observed. In addition, the RMSE reduces when a larger network input (more context frames) is used with a larger network. For instance, the RMSE with 5 total frames and 8000 hidden nodes is smaller than with 3 frames and 6000 hidden nodes; similarly, the RMSE with 5 frames and 10000 hidden nodes is smaller than with 3 frames and 8000 hidden nodes. It can thus be concluded that a larger network is needed to take full advantage of the temporal information added by using a bigger context window. Since the input weights in the ELM are kept fixed, the size of the hidden layer must also be increased in order to obtain the additional degrees of freedom needed to learn the added information.

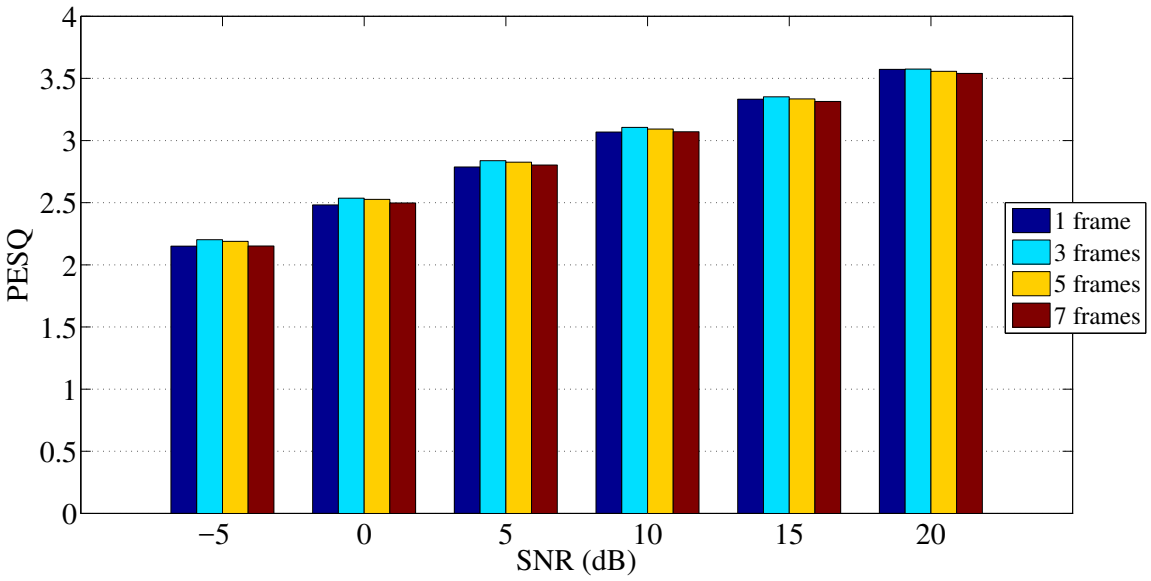


Figure 3.8: Average PESQ scores for a 7000 hidden node network with different context windows in matched noise tests.

Algorithm Comparison: Matched Noise

The performance of the OM-LSA algorithm and the ELM with matched noise is shown in Figure 3.10. The ELM outperforms the OM-LSA algorithm at all SNR levels with just a single hour training data. In addition, the performance margin increases as the average SNR reduces. The ELM is thus effective in suppressing of matched noise types when trained with very small datasets. Informal listening tests also showed the ELM-enhanced

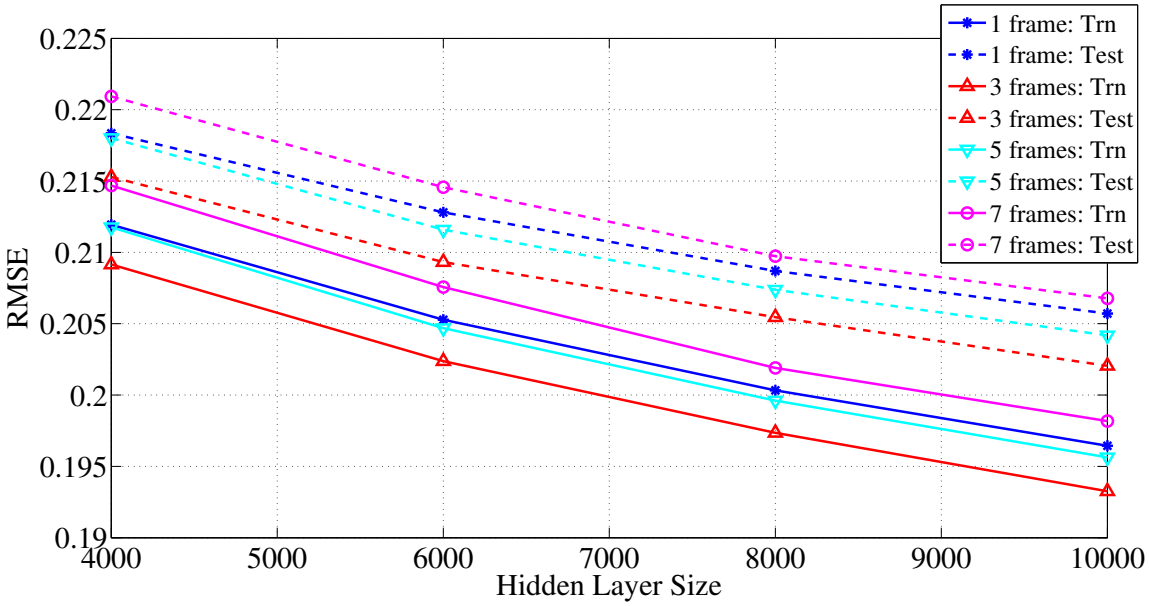


Figure 3.9: Training and testing error for networks of different sizes with different context windows.

speech was free of artifacts like musical noise.

Algorithm Comparison: Mismatched Noise

The performance of the OM-LSA algorithm and the ELM with mismatched noise is shown in Figure 3.11. Once again, the ELM outperforms the OM-LSA algorithm at all SNR levels with all of the training datasets. Unlike the OM-LSA estimator that fails to improve quality at the lowest SNR, the ELM always improves speech quality. The ELM is therefore able to learn a mapping between noisy spectral input and the target mask and effectively suppress added noise.

3.3.4 Comparison of ELM Architectures

The average PESQ scores for ELMs with IRM and log magnitude spectral targets [94] in matched noise are shown in Table 3.8. For a training set of any given length, the ELM with the IRM target outperforms the ELM with the log spectral target. The performance of the ELM with the IRM target, in particular, is much better with the smaller training datasets.

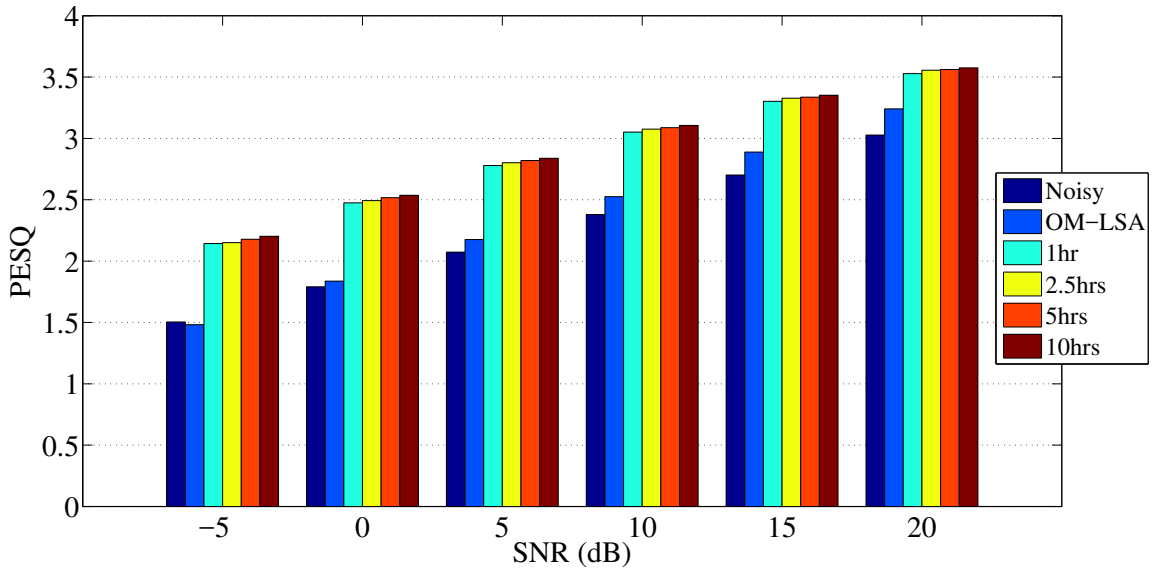


Figure 3.10: Average PESQ scores at different SNRs for OM-LSA and the ELM with matched training datasets of different sizes.

The log spectral ELM is unable to improve speech quality at 20dB SNR with 1 hour of training data, and it is outperformed by an average PESQ margin of 0.34 and 0.21 on the 1 and 2 hour datasets respectively. The IRM target is therefore more robust and generalizes better with small datasets. This could be because the IRM target, unlike the log spectral target, is bounded and therefore more easily fitted by the ELM. While the performance with both training targets is similar when trained with a 10 hour dataset, the IRM is still slightly superior and is consequently the better choice.

The average PESQ scores for ELMs with IRM and log magnitude spectral targets [94] in mismatched noise are shown in Table 3.9. The ELM with the IRM target with just 2.5 hours of training data is always as good, or better, than the performance of the ELM with the log spectral target. We can, therefore, conclude again that the IRM target is the better choice.

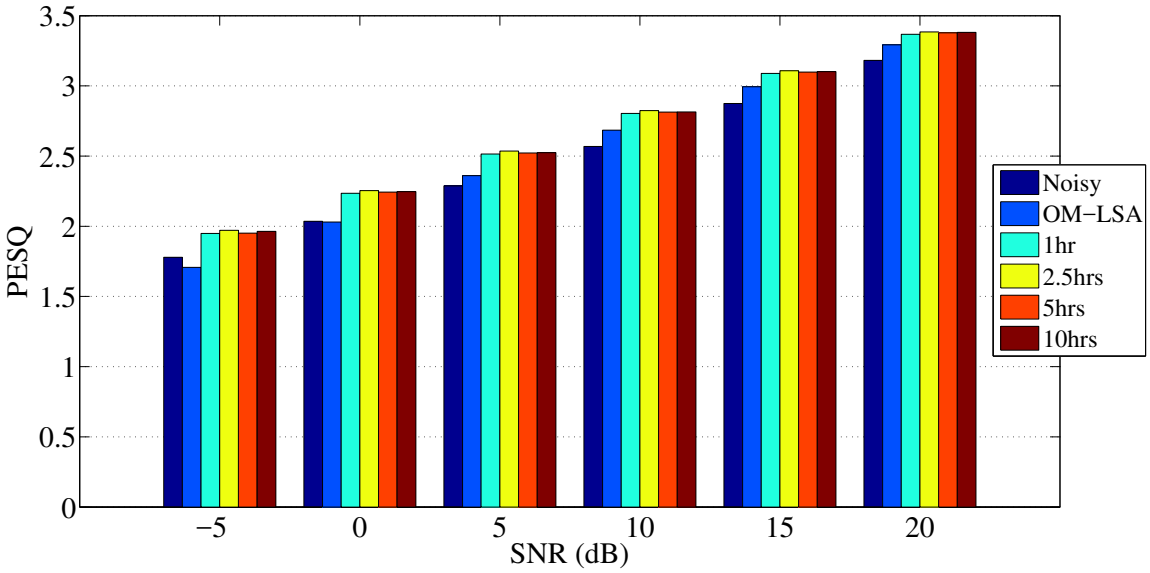


Figure 3.11: Average PESQ scores at different SNRs for the OM-LSA algorithm and the ELM with mismatched training datasets of different sizes.

3.4 Conclusions

In this chapter we developed and compared two speech enhancement frameworks based on the extreme learning machine. The frameworks used the original (batch) for small training datasets and the online-sequential ELM for larger datasets. The performance of the two frameworks was evaluated and compared to that of a leading MMSE algorithm, the OM-LSA estimator. In matched noise, the performance of the spectral mapping ELM was always superior to that of the OM-LSA algorithm with as few as 5 hours of training data. In mismatched noise, however, performance was approximately equivalent at higher SNR but superior at lower SNR values. The T-F framework was shown to be the superior framework. Its performance was almost always matched or exceeded that of the spectral mapping framework. Moreover, it was able to improve the quality of the noisy signal, even when the spectral mapping framework failed to do so. The performance of this framework was superior to the OM-LSA estimator with just one hour of training data in both matched and mismatched noise types. In addition, its performance was consistent over the wide range of training dataset sizes employed, thus showing good generalization with extremely

Table 3.8: Average PESQ scores for the ELM with different training targets and with matched training datasets of different sizes.

SNR (dB)	Noisy	Log Magnitude Spectrum				Ideal Ratio Mask			
		1hr	2.5hrs	5hrs	10hrs	1hr	2.5hrs	5hrs	10hrs
20	3.03	2.93	3.17	3.34	3.46	3.53	3.56	3.56	**3.57**
15	2.70	2.84	3.04	3.19	3.30	3.30	3.33	3.34	**3.35**
10	2.38	2.71	2.86	2.98	3.09	3.05	3.08	3.09	**3.11**
5	2.07	2.52	2.63	2.73	**2.84**	2.78	2.80	2.82	**2.84**
0	1.79	2.29	2.36	2.45	**2.54**	2.48	2.49	2.52	**2.54**
-5	1.50	2.00	2.04	2.12	**2.22**	2.14	2.15	2.18	2.20
AVE.	2.25	2.55	2.69	2.80	2.91	2.88	2.90	2.92	**2.93**

Table 3.9: Average PESQ scores for the ELM with different training targets and with mismatched training datasets of different sizes.

SNR (dB)	Noisy	Log Magnitude Spectrum				Ideal Ratio Mask			
		1hr	2.5hrs	5hrs	10hrs	1hr	2.5hrs	5hrs	10hrs
20	3.18	2.85	3.06	3.18	3.24	3.37	**3.38**	**3.38**	**3.38**
15	2.87	2.73	2.89	2.99	3.03	3.09	**3.11**	3.10	3.10
10	2.57	2.54	2.67	2.75	2.79	2.80	**2.82**	2.81	2.81
5	2.29	2.32	2.42	2.49	2.52	2.51	**2.54**	2.52	2.52
0	2.04	2.07	2.16	2.21	2.24	2.24	**2.25**	2.24	**2.25**
-5	1.78	1.82	1.89	1.93	**1.97**	1.95	**1.97**	1.95	1.96
AVE.	2.45	2.39	2.52	2.59	2.63	2.66	**2.68**	2.67	2.67

small training datasets.

CHAPTER 4
ON THE EXTREME LEARNING MACHINE AND MULTIVARIATE REGRESSION

4.1 Introduction

The ELM is described in the literature as an algorithm that is suitable for both multiclass classification and multivariate regression problems. In this chapter, we show that the ELM solution is not optimal for multivariate regression problems because it ignores correlations between the different response or target components. We propose two modifications to the ELM that account for the correlations between the elements of the response and yet adhere to the ELM ethos of learning without the use of iterative turning methods. We then compare the performance of the ELM and that of our proposed algorithms on several datasets. The rest of the chapter is organized as follows: in Section 4.2, we describe the problem of the ELM solution. In Section 4.3, we provide an overview of canonical correlation analysis and develop an improved multivariate ELM using canonical correlation analysis. We also propose an alternative model, the two-stage ELM in Section 4.4. Performance evaluations are presented in Section 4.5, computational considerations are discussed in Section 4.6, and conclusions are presented in Section 4.7.

4.2 The ELM with Multiple Outputs

The ELM algorithm, as discussed in Section 2.2, trains a neural network by randomly assigning values for input weights and biases, and solving for the output weights. Given a training dataset $\{(\mathbf{x}_i, \mathbf{t}_i), i = 1, \ldots, N\}$, the network is trained by finding output weights, $\hat{\boldsymbol{\beta}}$, such that

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{H}\boldsymbol{\beta} - \mathbf{T}||, \tag{4.1}$$

where $\mathbf{H}$ is the hidden node matrix, and $\mathbf{T}$ is the matrix of training targets. The output weights are thus obtained as

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T} \tag{4.2}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of the matrix $\mathbf{H}$. If the training targets are real-valued, the problem, (4.1), is a regression problem. If the training targets, on the other hand, are real-valued vectors, the problem is a multivariate regression problem.

Multi-target regression, also known multivariate or multi-output regression, refers to the task of predicting multiple continuous variables using a common set of input or predictor variables [102]. For multiple outputs or multivariate responses, each training target is a vector, $\mathbf{t}_i \in \mathbf{R}^m, i = 1, \ldots, N$. The target matrix, $\mathbf{T} \in \mathbf{R}^{N \times m}$ can be expressed as $\mathbf{T} = [\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_m]$, where $\mathbf{T}_j$, the j^{th} column of the matrix $\mathbf{T}$, is the j^{th} output variable or response. The least-squares solution for the output weights, (4.2), can be re-written as

$$\begin{aligned} [\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_2, \ldots, \hat{\boldsymbol{\beta}}_m] &= \mathbf{H}^{\dagger}\mathbf{T} \\ &= [\mathbf{H}^{\dagger}\mathbf{T}_1, \mathbf{H}^{\dagger}\mathbf{T}_2, \ldots, \mathbf{H}^{\dagger}\mathbf{T}_m]. \end{aligned} \tag{4.3}$$

The j^{th} column of parameters, $\hat{\boldsymbol{\beta}}_j$, therefore depends only on the j^{th} output, and the overall solution for the the regression parameters, $\hat{\boldsymbol{\beta}}_j, j = \{1, \ldots, m\}$ is comprised of the solutions to m independent univariate problems. In the case where the outputs are correlated, these correlations are not taken into account, and valuable information that can be used to improve the prediction accuracy of the model is not utilized [68, 79].

4.3 Improving the ELM with Canonical Correlation Analysis

4.3.1 Canonical Analysis

Canonical correlation analysis (CCA) studies that association or relationship between a set of predictor (independent) variables and a set of criterion (dependent) variables or between

two sets of variables [103]. Let $\mathbf{X} \in \mathbf{R}^{n \times p}$ and $\mathbf{Y} \in \mathbf{R}^{n \times q}$ be two multivariate datasets where $\mathbf{x}_i = [x_{i1}, \ldots, x_{ip}]$ and $\mathbf{y}_i = [y_{i1}, \ldots, y_{iq}]$ are corresponding pairs, and $i = 1, \ldots, n$, where n is the number of samples. The goal of canonical analysis is to seek a pair of linear transformations, $\mathbf{u} \in \mathbf{R}^q$ and $\mathbf{v} \in \mathbf{R}^p$, that maximizes the correlation between $\mathbf{Yu}$ and $\mathbf{Xv}$. In the most general form, the procedure seeks $K = \min(p, q)$ such pairs such that each subsequent pair maximizes the correlation subject to the constraint of being uncorrelated with any of the preceding pairs. Mathematically, this is expressed as [79]

$$(\mathbf{u}_k, \mathbf{v}_k) = \arg \max_{\substack{\{corr(\mathbf{Yu}, \mathbf{Yu}_l)=0\}_1^{k-1} \\ \{corr(\mathbf{Xv}, \mathbf{Xv}_l)=0\}_1^{k-1}}} corr(\mathbf{Yu}, \mathbf{Xv}). \tag{4.4}$$

The vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_K\}$ and $\{\mathbf{v}_1, \ldots, \mathbf{v}_K\}$ are referred to as the $\mathbf{y}$ and $\mathbf{x}$ canonical coordinates respectively, and the correlations

$$c_k = corr(\mathbf{Yu}_k, \mathbf{Xv}_k), \qquad k = 1, \ldots, K \tag{4.5}$$

are the canonical correlations. The canonical coordinates and correlations can be found by an eigenanalysis of the matrix [103]

$$\mathbf{Q} = \mathbf{R}_{YY}^{-1} \mathbf{R}_{YX} \mathbf{R}_{XX}^{-1} \mathbf{R}_{XY} \tag{4.6}$$

where $\mathbf{R}_{YY}$ and $\mathbf{R}_{XX}$ are the covariances matrices of $\mathbf{Y}$ and $\mathbf{X}$ respectively, and $\mathbf{R}_{XY} = \mathbf{R}_{YX}^T$ is the cross-covariance of the elements in $\mathbf{X}$ and $\mathbf{Y}$.

4.3.2 The Curds and Whey Procedure

Multivariate regression is an extension of multiple linear regression, where a single response is regressed on p predictors, to regressing $q > 1$ responses on p predictors. Given a dataset $D = \{(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^N, \mathbf{y}^N)\}$ where $\mathbf{x}^i = [x_{i1}, \ldots, x_{ip}]$ and $\mathbf{y}^i = [y_{i1}, \ldots, y_{iq}]$ are

respectively the predictors and responses, the multivariate regression model is given by

$$\mathbf{y} = \mathbf{x}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{4.7}$$

where $\boldsymbol{\beta}$ is a $q \times p$ matrix of regression coefficients, and $\boldsymbol{\epsilon}$ represents the errors. The least-squares estimates of the regression coefficients are obtained as [68]:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{4.8}$$

where $\mathbf{X} = [x_{ni}] \in \mathbf{R}^{N\times p}$ and $\mathbf{Y} = [y_{nj}] \in \mathbf{R}^{N\times q}$ are respectively the centered predictor matrix and response matrices. The k^{th} column of regression coefficients, $\hat{\boldsymbol{\beta}}_k$, depends on only the k^{th} column of the data matrix, $\mathbf{Y}_k$, hence the result is equivalent to performing separate ordinary least-squares (OLS) regressions for each of the q response variables.

The curds and whey (C & W) method is a shrinkage or regularization procedure for multiple linear regression with multivariate responses. The method aims to take advantage of the correlations between a set of multivariate responses, $\mathbf{y} = [y_1, \ldots, y_q]$, and obtain improved estimates, $\tilde{y}_i, i = 1, \ldots, q$, using a linear combination

$$\tilde{y}_i = \bar{y}_i + \sum_{k=1}^{q} b_{ik}(\hat{y}_k - \bar{y}_k) \qquad i = 1, \ldots, q \tag{4.9}$$

of the OLS estimates, $\hat{y}_i, i = 1, \ldots, q$, where $\bar{y}_i$ represents the sample mean of the i^{th} variable. The previous equation, (4.9) can be expressed in matrix form as

$$\tilde{\mathbf{y}} = \mathbf{B}\hat{\mathbf{y}} \tag{4.10}$$

where the vectors $\tilde{\mathbf{y}}$, $\hat{\mathbf{y}}$ are assumed to be (mean) centered. The C & W method uses canonical analysis to obtain an optimal shrinkage matrix $\mathbf{B}^* = \mathbf{U}^{-1}\mathbf{D}\mathbf{U}$ where $\mathbf{U} \in \mathbf{R}^{q\times q}$ is the canonical transformation matrix whose rows are the canonical coordinates of the

response and $\mathbf{D} = diag\{d_1, \ldots, d_q\}$ is a diagonal matrix. The elements of the matrix $\mathbf{B}^*$ are defined as

$$\{b_{ik}\}_{k=1}^{q} = \arg\min_{\{\beta_k\}_1^q} E\left[y_i - \sum_{k=1}^{q} \beta_k \hat{y}_k\right]^2, \; i = 1, \ldots, q, \tag{4.11}$$

where E is the expectation of the joint distribution of the data $(\mathbf{x}, \mathbf{y})$ to be predicted.

The optimal matrix, $\mathbf{B}^*$, results in a reduced mean-squared prediction error for each response

$$E[y_i - (\mathbf{B}^*\hat{\mathbf{y}})_i]^2 \leq E[y_i - \hat{y}_i]^2, \qquad i = 1, \ldots, q. \tag{4.12}$$

In practice, the elements of (4.11) are obtained by cross-validation [104]. Generalized cross-validation (GCV) can be used as a means of simplifying the computation [79], thus

$$\begin{aligned} \{b_{ik}\}_{k=1}^{q} = \arg\min_{\{\beta_k\}_1^q} \sum_{n=1}^{N} \left[y_{ni} - \sum_{k=1}^{q} \beta_k[(1-g)y_{nk} + g\hat{y}_{nk}]\right]^2, \\ i = 1, \ldots, q, \end{aligned} \tag{4.13}$$

where y_{ni} is the i^{th} component of the n^{th} data sample, $\hat{y}_{nk}$ is the k^{th} component of the n^{th} OLS prediction, and $g = \frac{1}{1-r}$ where $r = \frac{p}{N}$ is the ratio of the number of predictors variables to the number of training samples. Performing an eigenanalysis of the sample canonical correlation matrix gives

$$\begin{aligned} \widehat{\mathbf{Q}} &= (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \\ &= \widehat{\mathbf{U}}^{-1}\widehat{\mathbf{C}}^2\widehat{\mathbf{U}}, \quad \widehat{\mathbf{C}}^2 = diag\{\hat{c}_1^2, \ldots, \hat{c}_q^2\}, \end{aligned} \tag{4.14}$$

where $\widehat{\mathbf{U}}$ is a matrix whose rows are the sample $\mathbf{y}$-canonical coordinates and the elements of the matrix $\widehat{\mathbf{C}}^2$ are the respective sample squared canonical correlations.

The solution for the GCV shrinkage matrix is then obtained as

$$\mathbf{B} = \widehat{\mathbf{U}}^{-1}\widehat{\mathbf{D}}\widehat{\mathbf{U}}, \quad \widehat{\mathbf{D}} = diag\{\hat{d}_1, \ldots, \hat{d}_q\} \tag{4.15}$$

where

$$\hat{d}_i = \max\left\{\frac{(1-r)(\hat{c}_i^2 - r)}{(1-r)^2\hat{c}_i^2 + r^2(1-\hat{c}_i^2)}, 0\right\},\ i = 1, \ldots, q, \tag{4.16}$$

are the elements of the diagonal shrinkage matrix. The steps to implement the algorithm are as follows:

1. Transform each $\mathbf{y}$ to the canonical coordinate system, $\mathbf{y}' = \mathbf{U}\mathbf{y}$.

2. Perform a separate OLS regression for each of the transformed responses, $\mathbf{Y}'_i$, on the predictors, $\mathbf{X}$, obtaining $\hat{\mathbf{Y}}'_i, i = 1, \ldots, q$.

3. Separately scale (shrink) each $\mathbf{Y}'_i$ with the corresponding factor, d_i, obtaining $\tilde{\mathbf{Y}}' = \hat{\mathbf{Y}}\mathbf{D}$.

4. Transform back to the original $\mathbf{y}$-coordinate system, $\tilde{\mathbf{y}} = \mathbf{U}^{-1}\tilde{\mathbf{y}}'$.

4.3.3 The Canonical ELM

The canonical ELM (C-ELM) incorporates the C & W algorithm in the process of training an ELM. The procedure followed initially is identical to the ELM solution as proposed by G.B Huang *et al.* [71], however, it differs in the implementation of the least-squares solution for the output weights, (2.10). The procedure is summarized as follows:
given a training dataset $\{(\mathbf{x}_i, \mathbf{t}_i), i = 1, \ldots, N\}$ where $\mathbf{x}_i, \in \mathbf{R}^n$ and $\mathbf{t}_i, \in \mathbf{R}^m, m > 1$ are respectively the input data and target output vectors,

1. Randomly assign the input weights and hidden layer biases, $\{\mathbf{w}_i, b_i\}, i = 1, \ldots, L$.

2. Calculate the hidden layer output matrix $\mathbf{H}$, (2.7).

3. Center the hidden layer output matrix, $\mathbf{H}$, and the training targets matrix, $\mathbf{T}$ (2.8).

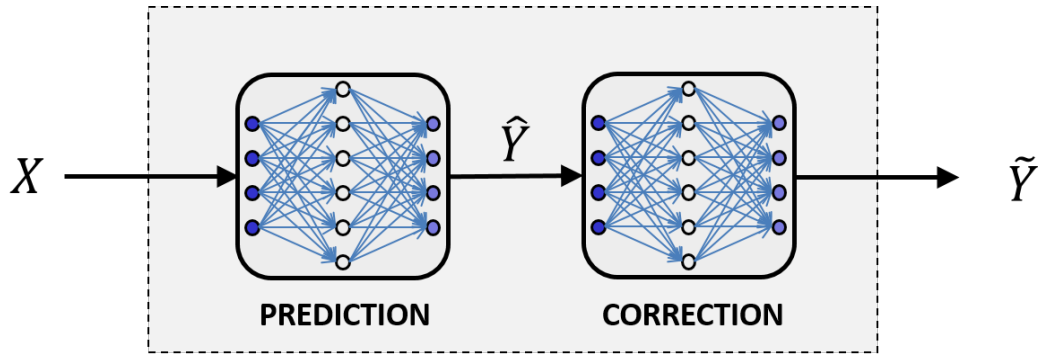


Figure 4.1: Block diagram of the two-stage ELM

4. Compute the sample canonical correlation matrix, $\widehat{\mathbf{Q}}$ matrix, (4.14), and perform an eigendecomposition of the matrix to obtain the sample canonical coordinates, $\widehat{\mathbf{U}}$, and correlations, $\widehat{\mathbf{C}}^2$.

5. Transform the target matrix, $\mathbf{T}$, to the canonical coordinate system by transforming each $\mathbf{t}$ as $\mathbf{t}' = \mathbf{U}\mathbf{t}$.

6. Compute the output weight matrix, $\hat{\boldsymbol{\beta}}$ (2.10), using the centered $\mathbf{H}$ matrix and the transformed target matrix, $\mathbf{T}'$, and then compute the predicted output, $\hat{\mathbf{T}}' = \mathbf{H}\widehat{\boldsymbol{\beta}}$

7. Shrink each $\widehat{\mathbf{T}}'_i$ with the corresponding factor, $d_i, i = 1, \ldots, q$.

8. Transform the predicted responses back to the original coordinate system, $\tilde{\mathbf{t}} = \mathbf{U}^{-1}\tilde{\mathbf{t}}'$

4.4 The Two-Stage ELM

The C & W approach to multivariate regression can be summarized as consisting of three operations, namely, transformation, prediction, correction (shrinkage), and (inverse) transformation. The middle stages, prediction and correction, involve learning a statistical data model and then compensating for the known inadequacies of the model. An alternate method could be to use a learning approach to accomplish the correction stage. The two-stage ELM (TS-ELM), which we introduce in this section, follows this paradigm.

A block diagram of the two-stage ELM is shown in Figure 4.1. The two-stage ELM consists of two neural networks in tandem. The first ELM is trained using the training dataset, $D = \{(\mathbf{x}^i, \mathbf{y}^i) : \mathbf{x}^i \in \mathbf{R}^p,\ \mathbf{y}^i \in \mathbf{R}^q, i = 1, \ldots, N\}$, and it therefore makes MMSE-optimal predictions of the output given the input features. The predictions of the first ELM are not used directly, instead they used to create a second dataset, $\widetilde{D} = \{(\hat{\mathbf{y}}^i, \mathbf{y}^i) : \hat{\mathbf{y}}^i, \mathbf{y}^i \in \mathbf{R}^q, i = 1, \ldots, N\}$, which is used to train the second ELM. The second ELM thus aims to improve the predictions of the first ELM by learning and applying the required shrinkage.

To obtain predictions for a new instance of the predictor vector, $\mathbf{x}^{new}$, the vector is input into the first ELM, and the output of the first ELM, $\hat{\mathbf{y}}^{new}$, is in turn input into the second ELM to produce the final output, $\tilde{\mathbf{y}}^{new}$. The TS-ELM thus replaces the algorithmic approach of the C & W method with a learning-based approach in which the correlations between the predicted responses are improved using a data-driven approach.

4.5 Performance Evaluation

To evaluate the performance of the proposed models, we conduct experiments using several benchmark datasets. First, we examine the characteristics of the algorithms using a synthetic dataset, then we compare the prediction accuracy of the algorithms on some real-world datasets.

4.5.1 Evaluation on a synthetic dataset

A synthetic dataset with 2000 training sample pairs, $(\mathbf{x}_i, \mathbf{y}_i)$, was created following [91] where each response, $\mathbf{y}_i = [y_{i1}, y_{i2}, \ldots, y_{i5}]$, is five-dimensional, and corresponding the corresponding predictor, $\mathbf{x}_i = [x_{i1}, x_{i2}]$, is two-dimensional. The dataset was generated as

follows:

$$
\begin{aligned}
y_{i1} &= 4\sin(x_{i1}) - 2\text{sinc}(x_{i2}) + 5 + n_{i1} \\
y_{i2} &= 3\sin(x_{i1}) - 3\cos(x_{i2}) + 2 + n_{i2} \\
y_{i3} &= -5\text{sinc}(x_{i1}) + 4\sin(x_{i2}) + 1 + n_{i3} \\
y_{i4} &= -2\sin(x_{i1}) - 2\sin(x_{i2}) - 5 + n_{i4} \\
y_{i5} &= 4\text{sinc}(x_{i1}) - 2\cos(x_{i2}) - 3 + n_{i5}
\end{aligned}
\tag{4.17}
$$

where $x_{i1} \sim N(0, 10)$, $x_{i2} \sim N(0, 5)$, and $n_{ij} \sim N(0, 0.5)$. The performance of ELM, C-ELM, and TS-ELM models was evaluated over a range of hidden layer sizes. For the TS-ELM, the hidden layer size of both networks, i.e. the first and second stage ELMs, was constrained to be the same. The performance results for each hidden layer size were obtained using 10-fold cross-validation, and all models used the same data folds.

The average training error of the models as the hidden layer size is varied is shown in Figure 4.2. While the training error of all the models initially decreases as the hidden layer size is increased, the ELM does not show noticeable improvement after 70 - 80 hidden nodes. The C-ELM and TS-ELM models, in contrast, are able to better fit the training data as the hidden layer size, and consequently, the number of available degrees of freedom, is increased.

The average test error is shown in Figure 4.3. The test error of the ELM and the TS-ELM can be seen to rapidly increase as the number of hidden nodes is increased. Such an increase in the test error can usually be attributed to overfitting of the training data; however, this is not the case for the ELM since there is no improvement in the training error as the hidden layer size is increased. The training error of the TS-ELM, in contrast, can be seen to continually decrease as the hidden layer size is increased. The ELM thus shows some degree of instability as the hidden layer size is increased. The test error of the C-ELM, on the other hand, decreases gradually as the hidden layer size is increased until

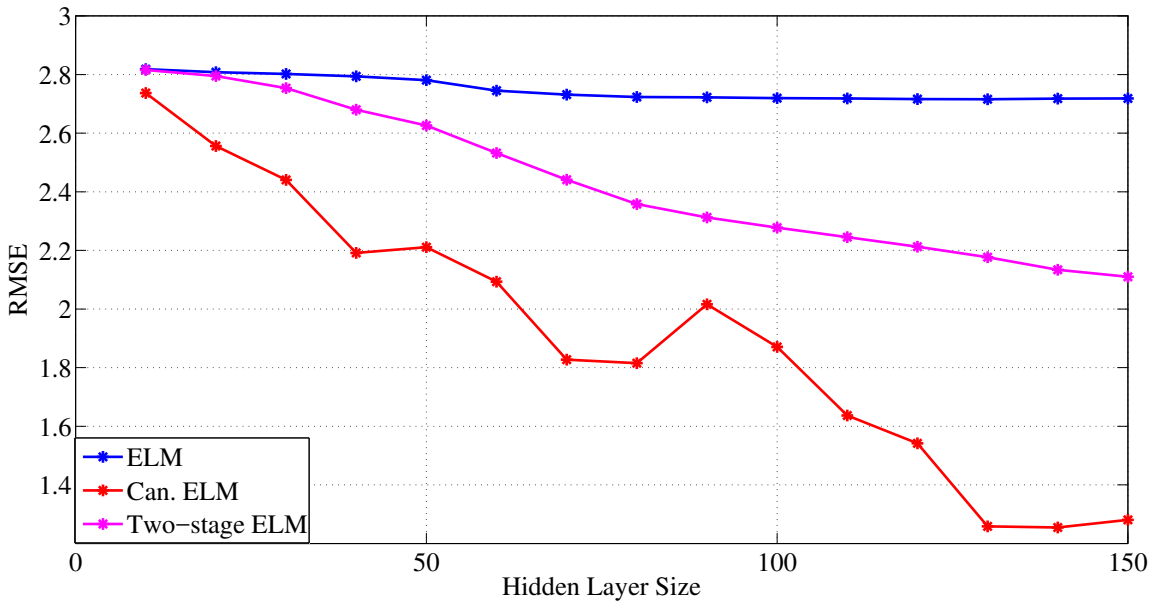


Figure 4.2: Average training error on the synthetic dataset for different hidden layer sizes.

the hidden layer size is about 130 nodes when it eventually increases. The difference in performance between the ELM and the C-ELM and TS-ELM models can thus be attributed to the presence of the correction stage in the latter two models.

Table 4.1: Performance of the models with highest prediction accuracy on the synthetic dataset. The standard deviation of the error is shown in parentheses.

Parameter	ELM	Canonical ELM	Two-stage ELM
Training RMSE	2.8185 (0.0045)	1.2583 (0.2580)	2.8158 (0.0047)
Testing RMSE	2.8344 (0.0393)	1.5364 (0.4241)	2.8504 (0.0471)
Hidden Layer Size	10	130	10

The parameters of the ELM, C-ELM, and TS-ELM models with the highest prediction accuracy are shown in Table 4.1. The smallest hidden layer size of 10 nodes is optimal for both the ELM and TS-ELM. A hidden layer size of 130 nodes, however, is optimal for the C-ELM. The TS-ELM has a marginally better training error than the ELM but also has a slightly worse prediction accuracy. This most likely reason that the TS-ELM does not

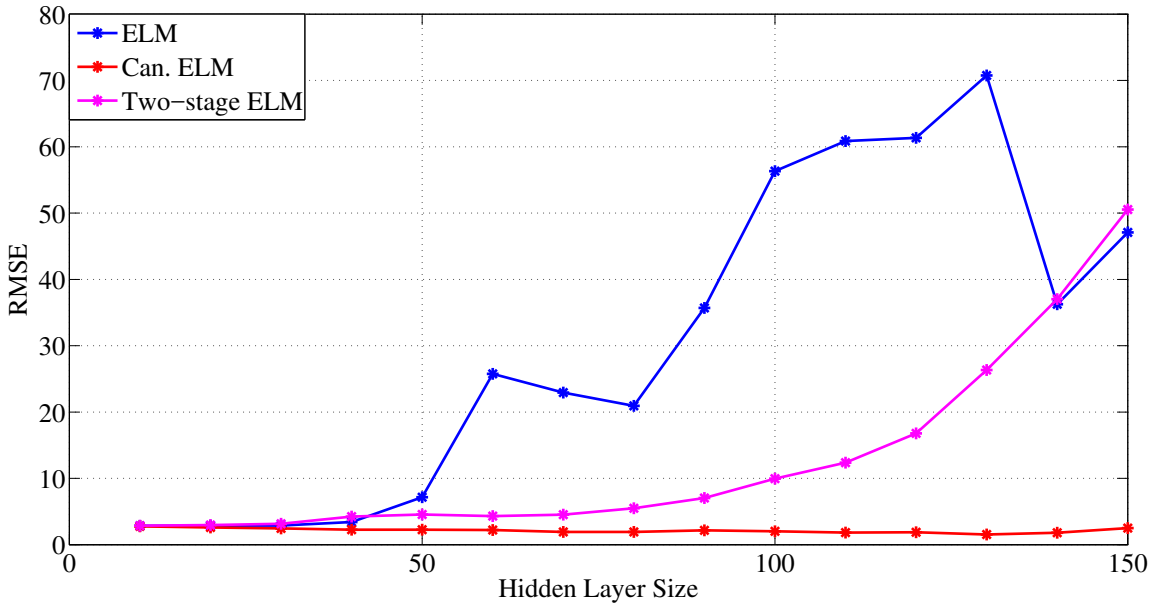


Figure 4.3: Average test error on the synthetic dataset for different hidden layer sizes.

show much improvement over the ELM despite the obvious potential displayed in Figure 4.2 and Figure 4.3 is because both the first and second stage ELMs in the TS-ELM were constrained to use the same hidden layer size. The C-ELM outperforms both the ELM and TS-ELM and its training and test errors are respectively about 55% and 45% lower than those of the ELM. The box plots of the training and test errors in Figure 4.4 and Figure 4.5 respectively show a greater variance in the performance of the C-ELM. The average error of the C-ELM, however, is always lower than that of the ELM, so it is a better choice for multivariate regression.

4.5.2 Evaluation on a real-world datasets

The real-world datasets used for the evaluation of the models are summarized in Table 4.2. The Slump dataset was obtained from the UCI machine learning repository [105], while the other datasets were obtained from the Mulan multi-target regression dataset repository [106]. A full description of the target and predictor attributes of the different datasets is contained in Appendix A.

The data were preprocessed by min-max normalizing of the predictors to the range [-

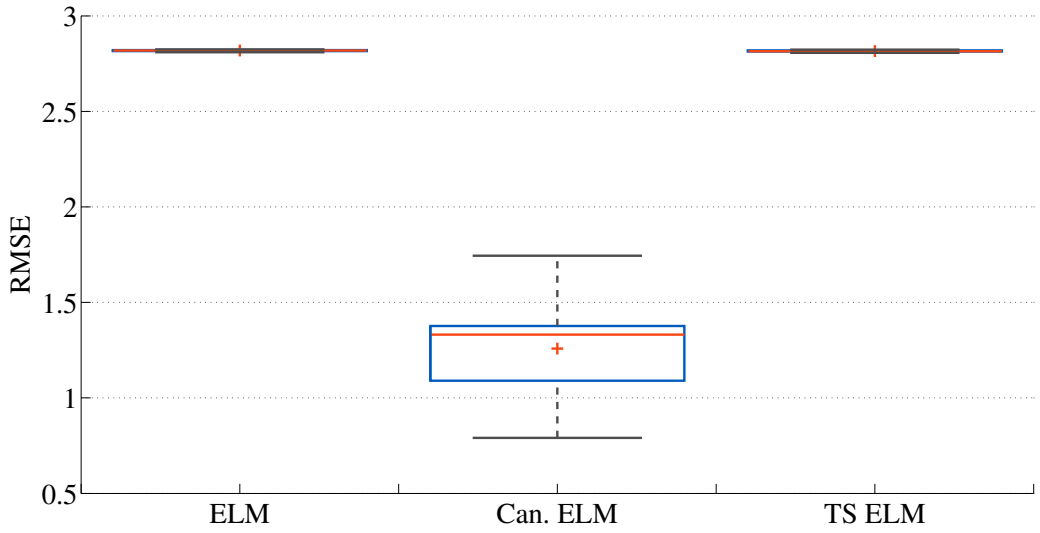


Figure 4.4: Box plot of the average training error for the models with optimal node sizes. The red '+' denotes the mean and the red line denotes the median error.

Table 4.2: Description of the real-world datasets.

Dataset	Samples (n)	Features (p)	Outputs (q)
WQ	1060	14	16
ATP1D	337	411	6
ATP7D	296	411	6
EDM	154	16	2
ENB	768	8	2
SCM1D	9803	280	16
SCM20D	8966	61	16
SLUMP	103	7	3

1,1] and normalizing the targets to the range [0,1]. All constant columns i.e. predictors that had the same value for all the training samples were removed from the data. This removal was necessary due to the min-max normalization that was utilized in preprocessing the data.

To determine the performance of each of the model, we split each dataset into 10 folds following standard cross-validation methodology. The training set for each fold was further subdivided into training and validation sets using k-fold cross-validation. The value of $k = 10$ was used for all the datasets except for the larger datasets, SCM1D and SCM20D,

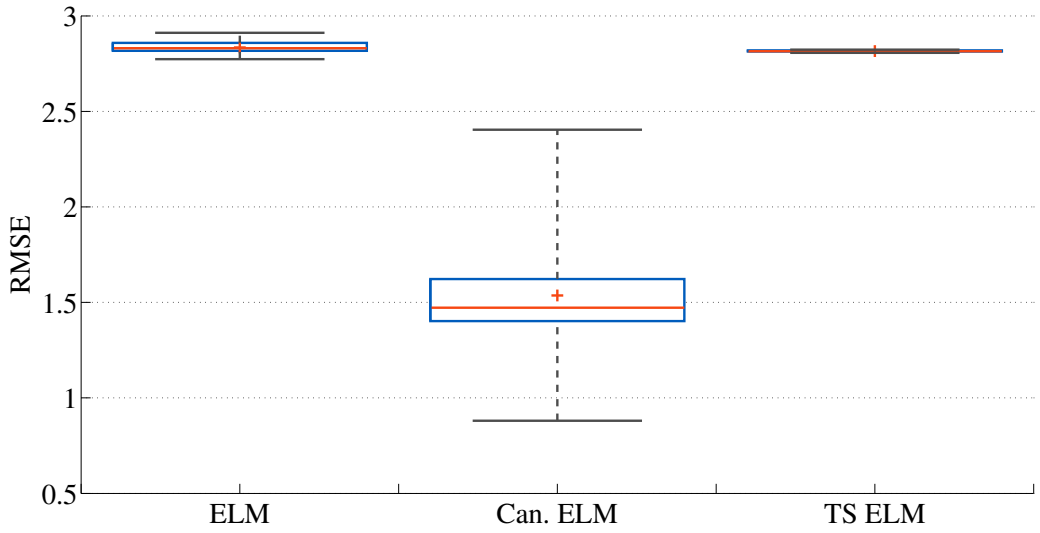


Figure 4.5: Box plot of the average prediction error for the models with optimal node sizes. The red '+' denotes the mean and the red line denotes the median error.

where a value of $k = 5$ was utilized.

The number of hidden layer nodes, the single hyper-parameter for both ELM and C-ELM, was optimized for each of the training data folds by averaging the prediction accuracy over the validation sets. A network with the optimal number of nodes was then re-trained on the entire training fold and the prediction accuracy of the model was determined using the corresponding test data fold. The TS-ELM, on the other hand, had two hyper-parameters, namely the hidden layer nodes in the first and second-stage ELM, to be optimized. The optimal parameters were determined by performing a grid search over all combinations of a selected range of hidden layer sizes in both networks. This was followed by the evaluation of the test error on the corresponding test data fold.

The performance of the models is summarized in Table 4.3. The C-ELM performs best on the WQ, ATP7D, EDM, ENB, and Slump datasets. The TS-ELM, on the other hand, performs best on the ATP1D and both SCM datasets. The prediction error of the C-ELM is between 1 - 13% lower than that of the ELM on the datasets on which it the optimal choice, while that of the TS-ELM is between 4 - 20% lower than that of the ELM when it is the best performer.

Table 4.3: Training and test errors (RMSE) on the real-world datasets. The standard deviation of the RMSE is shown in parentheses. **Bold** indicates the best result.

Dataset	ELM		Canonical ELM		Two-stage ELM	
	Train	Test	Train	Test	Train	Test
WQ	0.09394 (0.00186)	0.09840 (0.00694)	**0.09297** (0.00116)	**0.09708** (0.00697)	0.09424 (0.00184)	0.09795 (0.00792)
ATP1D	0.08963 (0.01139)	0.11176 (0.01077)	0.08550 (0.00554)	0.10727 (0.01110)	**0.08363** (0.00765)	**0.10642** (0.02158)
ATP7D	0.10617 (0.01757)	0.14905 (0.02808)	**0.09344** (0.01003)	**0.13181** (0.02862)	0.09619 (0.01211)	0.13375 (0.02730)
EDM	0.18062 (0.00941)	0.22269 (0.02999)	**0.17823** (0.01064)	**0.20512** (0.01655)	0.18142 (0.00474)	0.21558 (0.01990)
ENB	0.01906 (0.00281)	0.03810 (0.00907)	**0.01734** (0.00147)	**0.03304** (0.00670)	0.01938 (0.00268)	0.03623 (0.00506)
SCM1D	0.06044 (0.00253)	0.08145 (0.03411)	0.06006 (0.00411)	0.07067 (0.01402)	**0.05989** (0.00175)	**0.06504** (0.00250)
SCM20D	0.06519 (0.00043)	0.07081 (0.00164)	0.06514 (0.00046)	0.07033 (0.00149)	**0.06503** (0.00038)	**0.07001** (0.00163)
SLUMP	0.17538 (0.01472)	0.22009 (0.03144)	**0.16258** (0.02371)	**0.19317** (0.03125)	0.18153 (0.00766)	0.20316 (0.04116)

4.6 Computational considerations

The ELM is the least computationally intensive of all the models, while the TS-ELM in the most computationally intensive. This is because the C-ELM, in addition to the Moore-Penrose pseudoinverse, requires the computation and eigenanalysis of the sample canonical correlation matrix, (4.14), and the transformations of the target matrix and the OLS predictors as described in Section 4.3.2. The TS-ELM, on the other hand, requires the training of two ELM models. In addition, proper optimization of the hyper-parameters of the TS-ELM requires a grid search over a set of hidden layer sizes. This could prove to be prohibitive when either the hidden layer sizes of both networks are large or when the dataset is large, and discourage the use of the TS-ELM.

4.7 Conclusions

In this chapter, we examined the nature of the closed-form ELM solution for the output weights in a neural network and the performance of the ELM in multivariate regression tasks. The ELM solution was shown to be sub-optimal for multivariate regression since the weights for each dimension of the target, or output variable, are determined independently of the other target dimensions or output variables. As such, the correlations between the dimensions of the target, valuable information that can increase the accuracy of predictions, are ignored. Two methods for improving the ELM solution that adhered to the fundamental ELM design principle of learning without iterative tuning of weights were proposed. The canonical ELM, an algorithmic approach, had the highest prediction accuracy on a synthetic dataset with a prediction error 45% lower than that of the baseline ELM. The canonical ELM also had a prediction error that was 1% - 13% lower than that of the baseline ELM in tests on real-world datasets. The two-stage ELM, a data-driven approach, was also more accurate than the baseline ELM with a prediction error 4% - 20% than that of the baseline ELM on the real-world datasets.

CHAPTER 5

SPEECH ENHANCEMENT USING THE CANONICAL ELM

5.1 Introduction

In this chapter, we complete our development of improved extreme learning machine (ELM) algorithms by studying speech enhancement using the canonical ELM (C-ELM). We also extend our study of the ELM for speech enhancement by comparing the performance of the C-ELM and neural networks that are trained conventionally using stochastic gradient descent (SGD). The goal of this comparison is twofold: first, we determine if a key advantage of the ELM cited in the literature, namely, good generalization with small training sets, carries over to the multivariate regression speech enhancement problem. Second, we compare the performance characteristics of the ELM and the more established networks trained conventionally with SGD. The rest of the chapter is organized as follows: a system overview is given in the next section, and experiments are described in Section 5.3. The results of the ELM and C-ELM comparison are presented in Section 5.4, and the results of a comparison of the C-ELM and networks trained with SGD are presented in Section 5.5. Conclusion are presented in Section 5.6.

5.2 System Overview

A single hidden layer topology was used for the ELM, C-ELM, and SGD-trained networks. To facilitate comparisons with the results presented earlier in Chapter 3, the number of nodes in the hidden layer was fixed at 7000 nodes. Two training targets, namely, a log spectral target and the ideal ratio mask (IRM), (3.2), were utilized. As described previously in Chapter 3, both the batch ELM and OS-ELM algorithms were used in the training of the ELM network. The choice of algorithm was driven by the memory requirements of the

algorithms which, in turn, depended on the number of nodes in the hidden layer. The batch algorithm was used for networks with up to 6000 hidden nodes, and the OS-ELM algorithm was otherwise used.

The C-ELM network was trained with the canonical ELM algorithm described in Section 4.3. An online sequential variant of the canonical ELM algorithm was developed for training the C-ELM with datasets too large to fit in memory. Just like the OS-ELM discussed in Section 2.2.1, the method is based on the use of the recursive least-squares algorithm. The choice of algorithm, i.e. batch or sequential, followed the same concerns outlined for the choice of ELM algorithm. Consequently, the batch C-ELM algorithm was used for networks with up to 6000 hidden nodes, and the canonical OS-ELM algorithm was otherwise used.

The conventional network was trained by using the back-propagation algorithm to minimize a mean-square error criterion. Network parameters are updated using mini-batch stochastic gradient descent with momentum. The error criterion is

$$E = \frac{1}{N}\sum_{i=1}^{N} ||\hat{\mathbf{T}}_i(\mathbf{y}_i, \mathbf{\Theta}) - \mathbf{T}_i||^2 + \frac{\lambda}{2}||\mathbf{W}||_2^2 \tag{5.1}$$

where $\mathbf{y}_i$ is the input to the network, $\mathbf{\Theta} = \{\mathbf{W}, \mathbf{b}\}$, represents the weights and biases in the network, $\hat{\mathbf{T}}_i(\mathbf{y}_i, \mathbf{\Theta})$ is the output of the network, $\mathbf{T}_i$ is the desired training target, λ is the regularization coefficient, and N is the mini-batch size.

5.3 Experiments

All experiments were performed using recorded sentences from the IEEE Corpus included with the NOIZEUS database [97, 98]. The corpus is comprised of 72 lists, each list containing 10 sentences. Our noise samples came from a database of 100 non-speech sounds [99]. Both the noise-free speech and noise recordings were resampled to 8kHz. The training datasets wee comprised of an appropriate number of sentences from lists 1 - 57 for the

Table 5.1: Description of noise types.

Noise	Description	Noise	Description
n1	Crowd	n6	Water
n2	Machine	n7	Wind
n3	Alarm/Siren	n8	Bell
n4	Traffic/Car	n9	Cough
n5	Animal	n10	Clap

desired training set size, while testing was done with the 50 sentences from lists 68 - 72. Noisy utterances were created by adding 10 types of noise to each of the sentences at six noise levels ranging from 20dB to -5dB in 5dB steps. The noise types are listed in Table 5.1.

Short-time Fourier analysis was done using a Hamming window, 32ms frames, with 50% overlap, and spectral features extracted from the clean speech, noisy speech, and from the added noise signals were used to create input-output pairs for training the networks. The ELM and C-ELM networks used log magnitude spectral input features which, following previous results, were normalized to the range of [-1,1]. The training targets, the log magnitude spectral target or the IRM target, were not normalized.

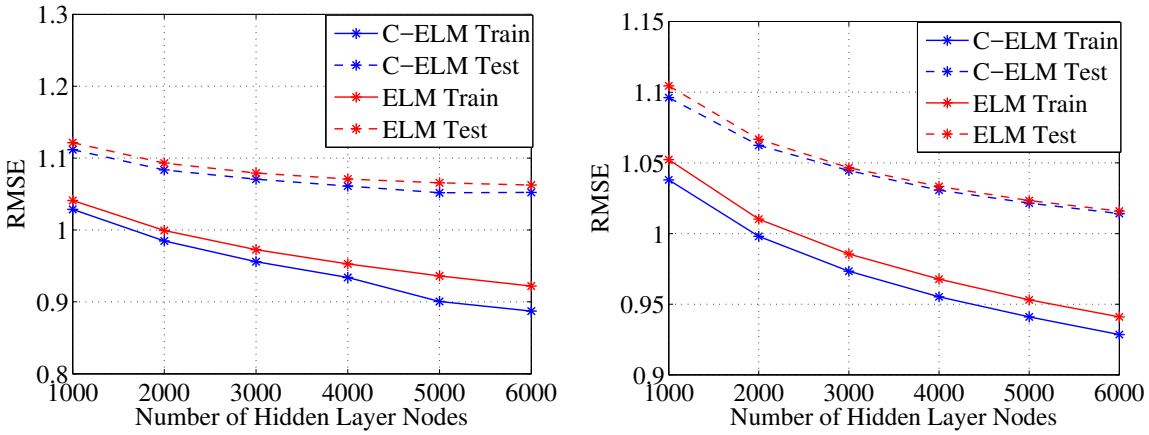
The conventional, SGD-trained network, in following common practice, used log power spectral input/output features. Input features were normalized to have zero mean and unit variance, while training targets were not normalized. The hidden layer of the conventional network used the rectified linear unit (ReLU) activation functions [107, 108]. Weights and biases of all the layers were initialized following the method of He *et al*. [109], and the networks were trained using gradient descent with momentum. The initial learning rate was set to 0.0001 for the first 10 epochs, then decreased by 10% every subsequent 10 epochs, and the the momentum coefficient was set for 0.9. A mini-batch size of 128 samples was used, and the networks were trained for 30 epochs. All networks were implemented and trained using the TensorFlow library [110].

To allow the networks to take advantage of temporal information, each input vector included adjacent time frames. Consequently, each input vector was constructed as

$$\mathbf{y}_i = [\mathbf{x}_{i-l}, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_{i+l}]. \tag{5.2}$$

As was done previously, zero to five context frames, i.e. $l = [0, 5]$, for an input length of up to eleven frames, were used in training and evaluation of the enhancement systems. The performance of the systems was tested with matched and mismatched noise types using the same noise categories in Table 5.1, and results were objectively evaluated using perceptual evaluation of speech quality (PESQ), a standard perceptual quality measure that has been shown have high correlation with subjective test scores [97, 101]. PESQ scores range between -0.5 to 4.5, with higher scores corresponding to higher perceptual speech quality.

5.4 A Comparison of the ELM and Canonical ELM



(a) 1 hour (b) 2.5 hours

Figure 5.1: Training and prediction error for the ELM and C-ELM with log magnitude spectral targets and different amounts of training data.

The training and prediction errors for for the ELM and C-ELM with different amounts of training data is shown in Figure 5.1. The training error of the C-ELM is smaller than the training error of the ELM when either 1 or 2.5 hours of speech data is used for training

the networks. With 1 hour of training data, the difference in the magnitude of the training errors appears to increase as the hidden layer size is increased suggesting that the C-ELM is increasingly effective at learning with the smaller dataset as the number of hidden nodes, and consequently, degrees of freedom, are increased. This is similar to the observation with the synthetic dataset in Figure 4.2 where the training error of the ELM remained constant as the hidden layer size was increased but that of the C-ELM continued to decrease.

The prediction error of the C-ELM is also smaller than that of the ELM when training is performed with either of the training sets. With 1 hour of training data, the difference in prediction error of the models is about constant, irrespective of the hidden layer size, however, with 2.5 hours of training data, the difference in the prediction errors of the models becomes negligible as the hidden layer size increases.

Table 5.2: Average PESQ scores for the ELM and C-ELM with training datasets of different sizes in matched noise tests.

SNR (dB)	Noisy	ELM: Log Magnitude Spectrum				C-ELM: Log Magnitude Spectrum			
		1hr	2.5hrs	5hrs	10hrs	1hr	2.5hrs	5hrs	10hrs
20	3.03	2.93	3.17	3.34	3.46	2.97	3.20	3.36	3.44
15	2.70	2.84	3.04	3.19	3.30	2.88	3.07	3.20	3.27
10	2.38	2.71	2.86	2.98	3.09	2.75	2.89	3.00	3.06
5	2.07	2.52	2.63	2.73	2.84	2.56	2.67	2.76	2.81
0	1.79	2.29	2.36	2.45	2.54	2.33	2.40	2.48	2.52
-5	1.50	2.00	2.04	2.12	2.22	2.05	2.09	2.16	2.20
AVE.	2.25	2.55	2.69	2.80	2.91	2.59	2.72	2.83	2.88

The average PESQ scores for the ELM and C-ELM with a log magnitude spectral target in matched and mismatched noise tests are shown in Table 5.2 and Table 5.3 respectively. It can be seen from Table 5.2 that in matched noise, the C-ELM performs better than the ELM at all SNR levels and on average with the 1, 2.5, and 5 hour training sets, however, the ELM is performs slightly better with the 10 hour training set. The difference in average PESQ scores for the different training datasets, 0.04 with 1 hour of training data, and a smaller 0.03 with 2.5 and 5 hours of training data, is reflective of the smaller difference in

Table 5.3: Average PESQ scores for the ELM and C-ELM with training datasets of different sizes in mismatched noise tests.

SNR (dB)	Noisy	ELM: Log Magnitude Spectrum				C-ELM: Log Magnitude Spectrum			
		1hr	2.5hrs	5hrs	10hrs	1hr	2.5hrs	5hrs	10hrs
20	3.18	2.85	3.06	3.18	3.24	2.89	3.10	3.22	3.28
15	2.87	2.73	2.89	2.99	3.03	2.76	2.93	3.02	3.07
10	2.57	2.54	2.67	2.75	2.79	2.58	2.72	2.79	2.83
5	2.29	2.32	2.42	2.49	2.52	2.36	2.47	2.53	2.57
0	2.04	2.07	2.16	2.21	2.24	2.13	2.21	2.26	2.30
-5	1.78	1.82	1.89	1.93	1.97	1.90	1.96	2.00	2.03
AVE.	2.45	2.39	2.52	2.59	2.63	2.44	2.57	2.64	2.68

prediction error observed with the larger 2.5 hour training set in Figure 5.1(b).

In mismatched noise tests, Table 5.3, the C-ELM performs better than the ELM at all SNR levels irrespective of the training dataset, and, consequently, on average. The difference of 0.05 in PESQ scores averaged over all SNR levels is the same for all of the different training datasets.

Table 5.4: Average PESQ scores for the ELM and C-ELM with training datasets of different sizes in matched noise tests.

SNR (dB)	Noisy	ELM: Ideal Ratio Mask				C-ELM: Ideal Ratio Mask			
		1hr	2.5hrs	5hrs	10hrs	1hr	2.5hrs	5hrs	10hrs
20	3.03	3.53	3.56	3.56	3.57	3.53	3.55	3.55	3.56
15	2.70	3.30	3.33	3.34	3.35	3.30	3.32	3.32	3.33
10	2.38	3.05	3.08	3.09	3.11	3.05	3.06	3.07	3.08
5	2.07	2.78	2.80	2.82	2.84	2.77	2.79	2.80	2.81
0	1.79	2.48	2.49	2.52	2.54	2.47	2.48	2.49	2.51
-5	1.50	2.14	2.15	2.18	2.20	2.13	2.14	2.16	2.18
AVE.	2.25	2.88	2.90	2.92	2.93	2.87	2.89	2.90	2.91

The average PESQ scores for the ELM and C-ELM with an IRM target in matched and mismatched noise tests are shown in Table 5.4 and Table 5.5 respectively. In matched noise tests, Table 5.4, the average scores of the ELM and C-ELM are identical at the higher SNR levels with 1 hour of training data, however, the ELM performs marginally better at the

Table 5.5: Average PESQ scores for the ELM and C-ELM with training datasets of different sizes in mismatched noise tests.

SNR (dB)	Noisy	ELM: Ideal Ratio Mask				C-ELM: Ideal Ratio Mask			
		1hr	2.5hrs	5hrs	10hrs	1hr	2.5hrs	5hrs	10hrs
20	3.18	3.37	3.38	3.38	3.38	3.37	3.38	3.38	3.38
15	2.87	3.09	3.11	3.10	3.10	3.09	3.11	3.10	3.10
10	2.57	2.80	2.82	2.81	2.81	2.80	2.82	2.81	2.81
5	2.29	2.51	2.54	2.52	2.52	2.51	2.53	2.52	2.52
0	2.04	2.24	2.25	2.24	2.25	2.24	2.25	2.24	2.24
-5	1.78	1.95	1.97	1.95	1.96	1.95	1.97	1.95	1.96
AVE.	2.45	2.66	2.68	2.67	2.67	2.66	2.68	2.67	2.67

lower SNR levels. The difference of 0.01 in the PESQ scores at these SNR levels and on average is, however, negligible. The ELM also performs marginally better than the C-ELM with the 2.5, 5, and 10 hour training datasets. Once again, the difference in PESQ scores which ranges from 0.01 - 0.02 is insignificant.

In mismatched noise tests, Table 5.5, the PESQ scores of the ELM and C-ELM are identical for all the training datasets and at all SNR levels except for two cases, 5dB with 2.5 hours and 0dB with 10 hours of training data, where there is a negligible difference of 0.01. The PESQ scores for the different training datasets averaged over all SNR levels are identical for both ELM and C-ELM.

The performance results of the ELM and C-ELM with both the spectral and IRM targets show that not much of an advantage, in general, is gained from choosing the C-ELM instead of the ELM for speech enhancement. Although the C-ELM performs better than the ELM in some cases, the improvement, as evidenced by the PESQ scores, is too small to be perceptually noticeable. In addition, the C-ELM is more computationally expensive than the ELM. As such, very little benefit is gained from incurring the additional cost of computation, and the ELM with the IRM target remains the best choice.

5.4.1 Discussion of Results

The C-ELM outperformed the ELM on both the synthetic and real-world datasets in Section 4.5, however, it did not present a clear advantage when used for speech enhancement. Two factors are likely responsible for these observations. The first factor, random initialization of the input weights in the ELM, could be responsible for the slightly worse performance of the C-ELM with the lRM target, and the C-ELM spectral target and 10 hours of training data. The random initialization of the ELM is known to be sub-optimal as a poor set of weights could result in poor performance [111, 112]. The random initialization also affects the correlations of the hidden layer activations and is the most likely cause of the larger variance in the training and prediction errors of the C-ELM as observed in Figures 4.4 and 4.5.

The second factor is the reduced shrinkage of the Curds & Whey algorithm as the sample size or amount of training data becomes larger. As previously described in Section consisting of three operations, namely, transformation, prediction, correction (shrinkage), and (inverse) transformation. The elements of the GCV shrinkage matrix were obtained in (4.16) as

$$\hat{d}_i = \max\left\{\frac{(1-r)(\hat{c}_i^2 - r)}{(1-r)^2\hat{c}_i^2 + r^2(1-\hat{c}_i^2)}, 0\right\},\ i = 1, \ldots, q, \tag{5.3}$$

where $\hat{c}_i^2$ are the sample squared canonical correlations and $r = \frac{p}{N}$ is the ratio of the number of predictors to the sample size. Figure 5.2 illustrates the value of the shrinkage factors, d_i (5.3), as a function of the squared canonical correlations, c_i^2, for various values of the ratio r. As the amount of training data increases, r becomes small, and the shrinkage factors, $d_i, 1 = 1, \ldots, q$ approach 1 for all values of the squared canonical correlation, c_i^2. Consequently, there is very little shrinkage, and the method is ineffective. The value of the sample to predictor ratio, r, for the different datasets is shown in Figure 5.3.

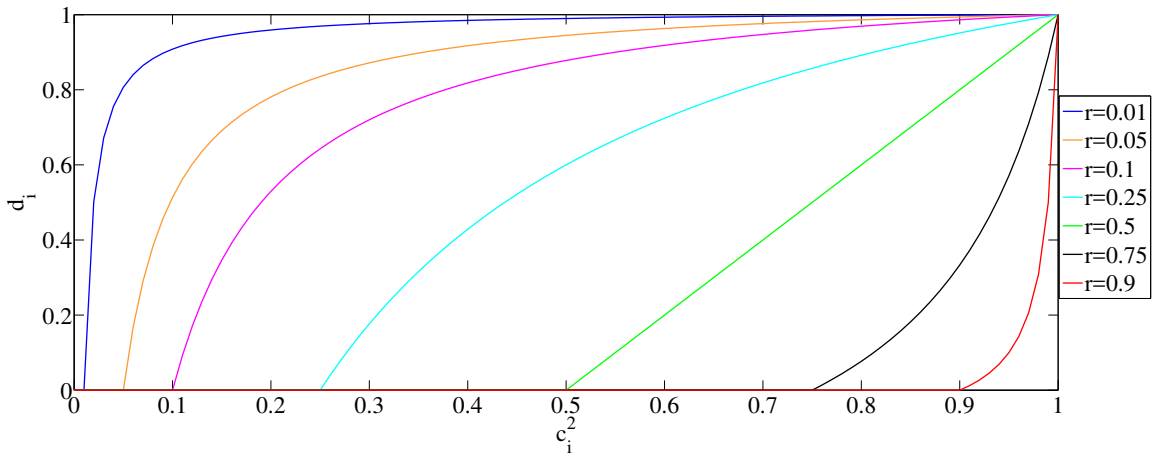


Figure 5.2: Values of the GCV shrinkage factor for various values of the ratio r.

5.5 A Comparison of the Canonical ELM and Conventionally Trained Networks

5.5.1 Effect of Acoustic Context

The average PESQ scores on a matched noise test set when using input features from context windows of different sizes are shown in Figure 5.4. The results for the C-ELM with the spectral target in Figure 5.4(a) are virtually identical to those of the ELM with the same spectral target shown in Figure 3.3: expanding the input with a single context frame, or an input of 3 frames in total, improves performance, but further expansion of the input degrades performance. The results of the SGD-trained network, on the other hand, show that for higher-SNR signals, the use of up to two additional frames (a total input of 5 frames) improves performance, however, any further expansion, degrades performance. The decline in performance is, however, smaller than was seen with the C-ELM. For speech signals with an SNR of 5dB or lower, performance continually improves as the size of the context window is increased.

The average PESQ scores on a matched noise test set when using input features from context windows of different sizes and an IRM target ARE shown in Figure 5.5. The performance of the C-ELM with the IRM target is once again virtually identical to that of the ELM with the IRM target shown in Figure 3.8: performance is improved with the use of

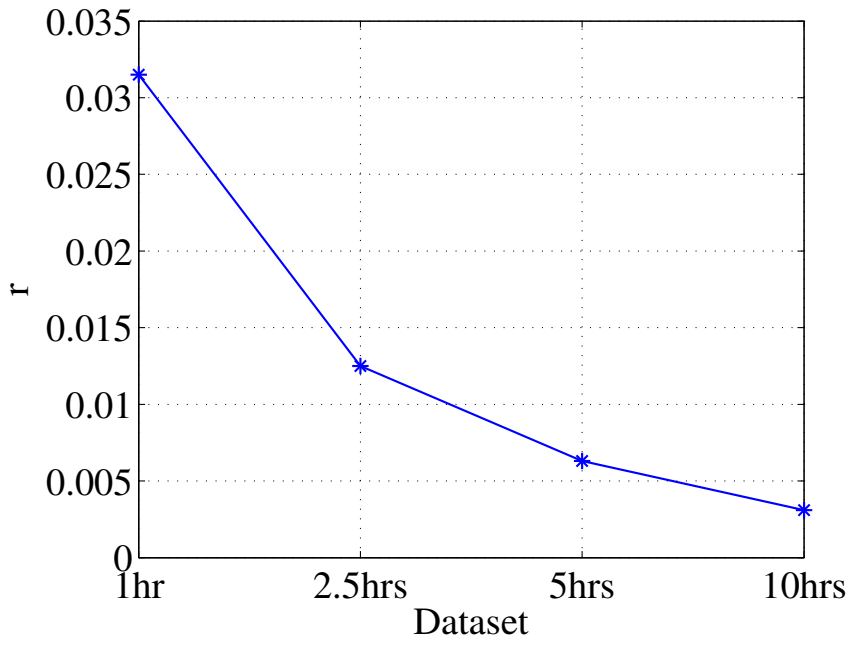


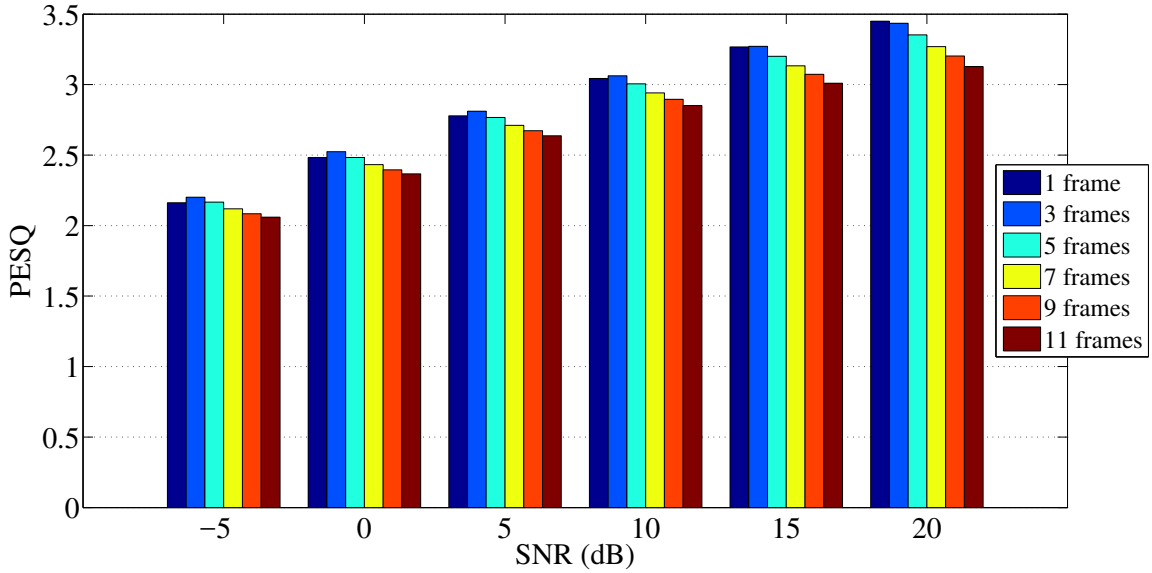
Figure 5.3: Predictor to sample size ratio for the C-ELM training datasets.

an additional frame and further expansion of the input degrades performance. The decline in performance is, however, marginal, unlike that which was seen with the C-ELM that had the spectral target. The results of the SGD-trained network, on the other hand, show that for higher-SNR signals, the use of up to two additional context frames (a total input of 5 frames) improves performance. Further expansion of the input causes a slight decline in performance. For lower-SNR signals, the use of more than two context frames does not degrade performance, but it also does not further improve the performance.

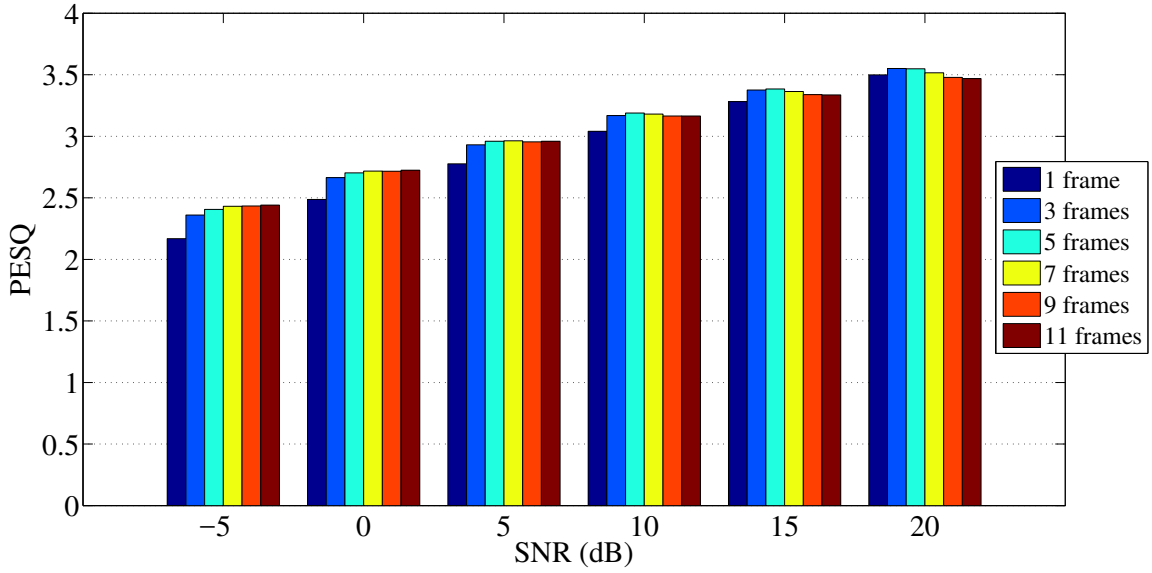
The SGD-trained network which has both input and output weights tuned iteratively is, therefore, better able to use the additional information provided by the context window. As was explained in Section 3.3, the size of the hidden layer in the ELM must be increased in order to take full advantage of the additional information.

5.5.2 Effect of Training Set Size

The performance of the C-ELM and SGD-trained networks in matched noise tests is shown in Figure 5.6. The networks use either a log spectral (LFFT) target or an ideal ratio mask

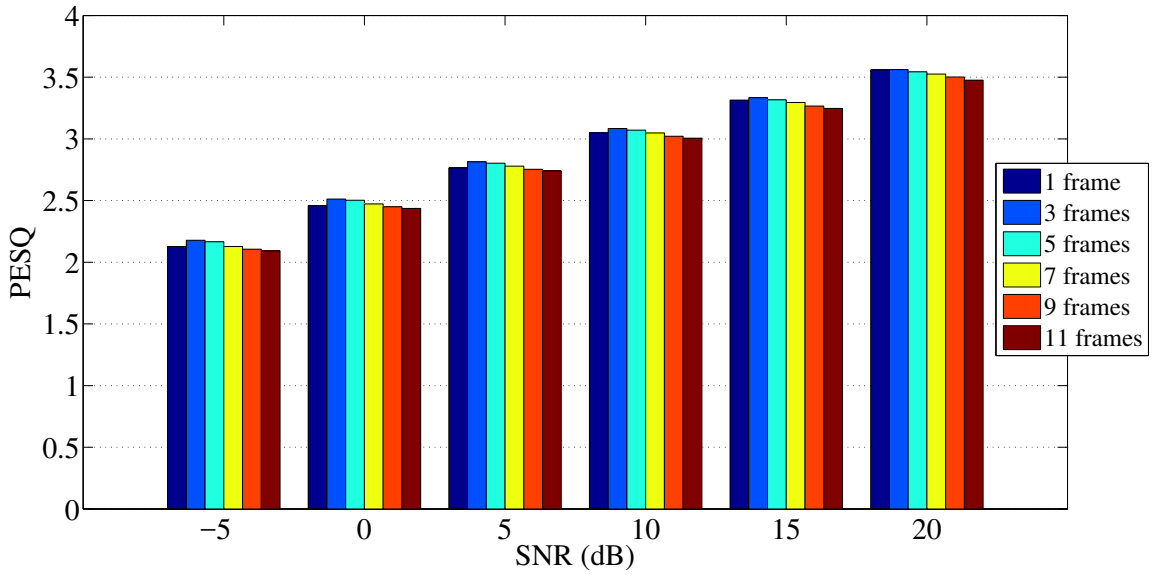


(a) C-ELM

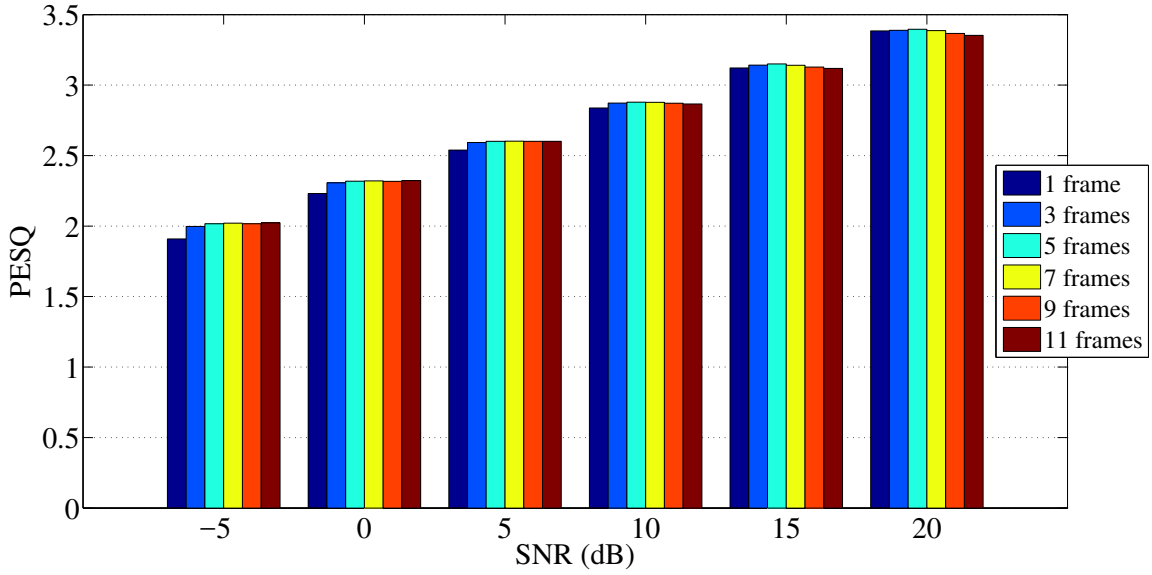


(b) SGD Network

Figure 5.4: Average PESQ scores for a 7000 node single-hidden layer network with different context windows in matched noise tests. The networks used a log spectral target and were trained with 10 hours of speech data.



(a) C-ELM



(b) SGD Network

Figure 5.5: Average PESQ scores for a 7000 node single-hidden layer network with different context windows in matched noise tests. The networks used an IRM target and were trained with 10 hours of speech data.

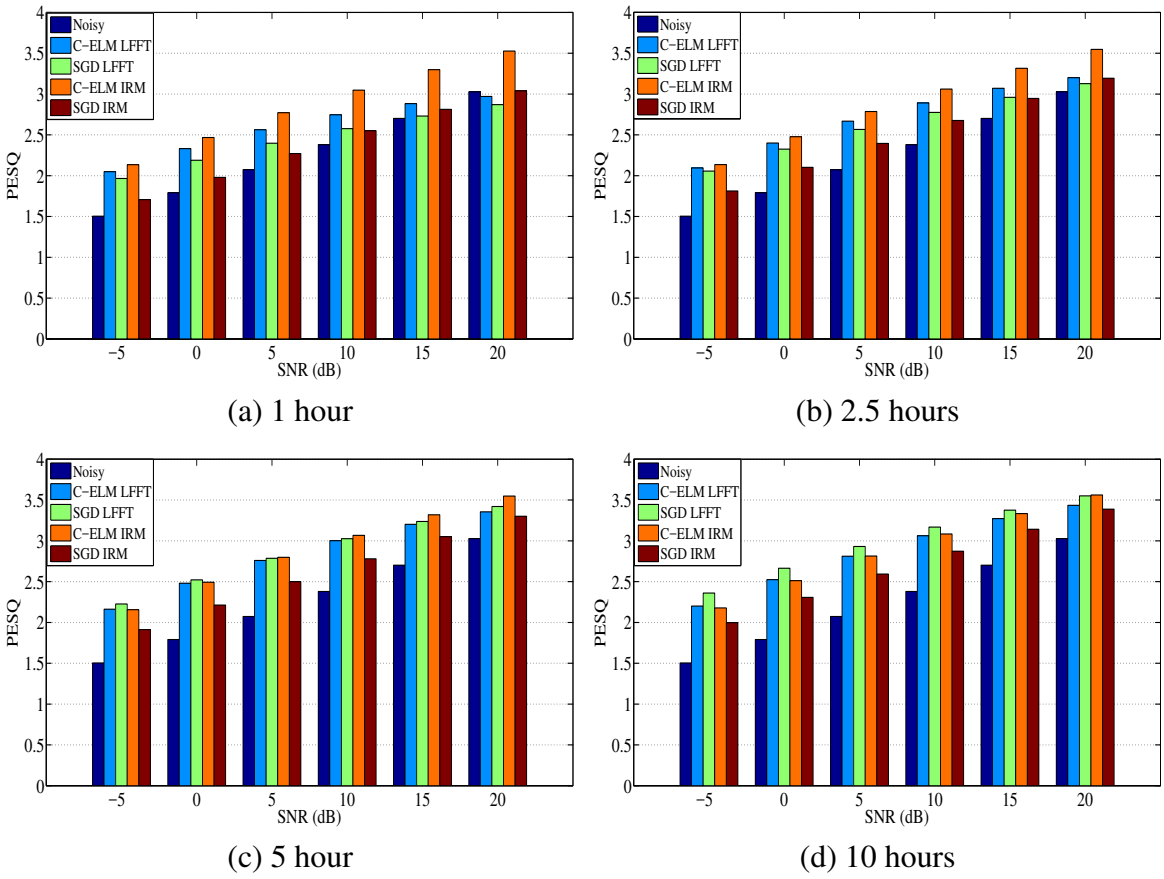


Figure 5.6: Average PESQ scores for the C-ELM and SGD-trained networks with different training targets and datasets of various sizes in matched noise tests. All networks used a single hidden layer with 7000 nodes and 3 frame input.

(IRM) target, and the size of the training datasets ranged from approximately 1 hour to 10 hours in length. As previous mentioned in Section [CITE], the ELMs used magnitude spectral features while the SGD-networks used power spectral features. From the figure, it can be seen that the C-ELM networks outperform the SGD-networks at all SNR levels on the smaller 1 hour and 2.5 hour datasets. The relative performance of the SGD-networks, particularly the SGD network with the log spectral target, SGD LFFT, improves beginning with the 5 hour training dataset. Although C-ELM IRM performs better than SGD LFFT at higher SNR levels of 5dB and above, SGD LFFT is margnially better at lower SNR levels. The same trend in also seen with the 10 hour training dataset. Notably, the SGD-trained network with the IRM target, SGD IRM, does not perform as well as C-ELM IRM with

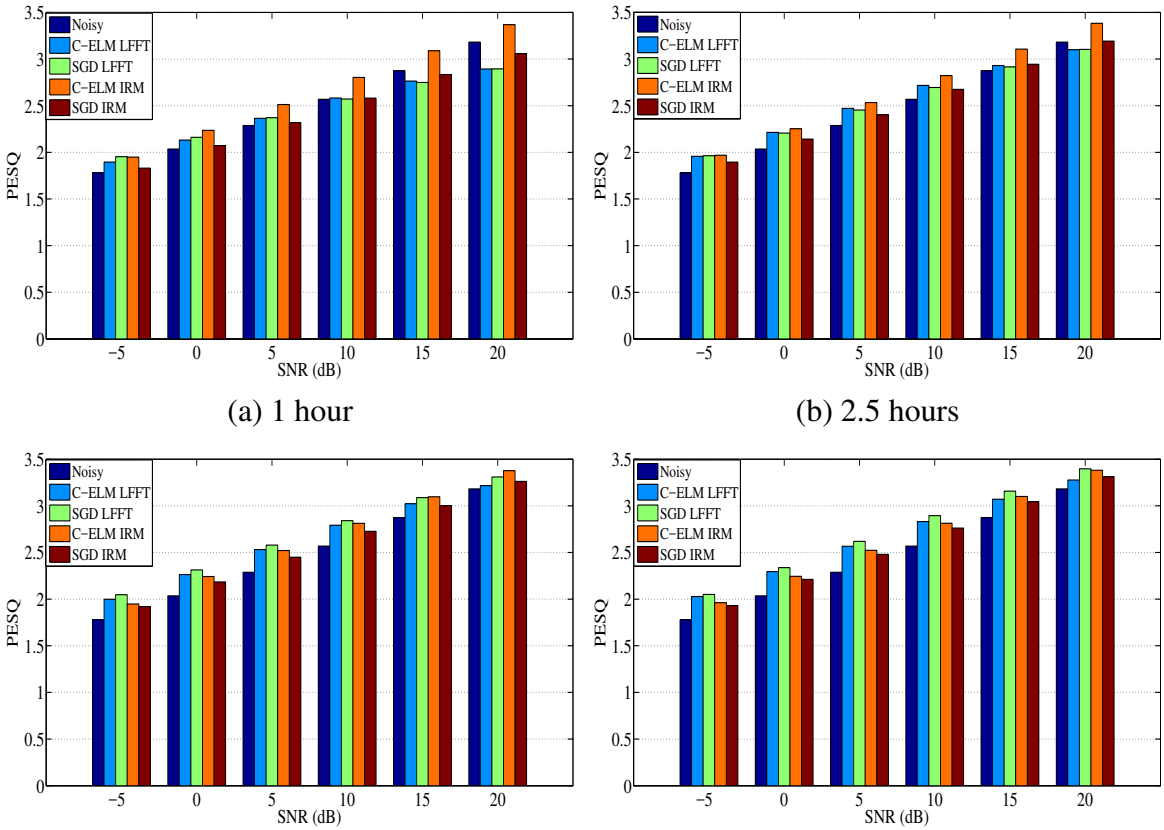


Figure 5.7: Average PESQ scores for the C-ELM and SGD-trained networks with different training targets and datasets of various sizes in mismatched noise tests. All networks used a single hidden layer with 7000 nodes and 3 frame input.

any of the training datasets.

The performance of the C-ELM and SGD-trained networks in mismatched noise tests is shown in Figure 5.7. The performance trends with the smaller 1 hour and 2.5 hour datasets are similar to those observed in the matched noise tests, however, this time, the performance of the SGD LFFT network is about equivalent to that of the C-ELM IRM network at the lower SNR levels. With 5 hours of training data, the SGD LFFT network begins to outperform the C-ELM networks SNR levels of 10dB and below, and it outperforms the C-ELM networks at all SNR levels with 10 hours of training data. Once again, the SGD IRM network does not perform as well as the C-ELM IRM network with any of the training datasets.

The C-ELM can thus be seen to be particularly effective at learning from small datasets. The performance of the C-ELM IRM, in particular, is notable when the closeness of the average PESQ scores in Tables 5.4 and 5.5 is considered: the C-ELM IRM network trained with just 1 hour of data performs almost as well as the SGD LFFT network trained with 10 hours data.

5.6 Conclusion

In this chapter, we examined the use of the canonical ELM (C-ELM) for speech enhancement. Objective test results showed that the performance of the C-ELM with a spectral target was superior to that of the ELM with a spectral target when training was carried out with the smaller datasets; however, the performance of the ELM with the IRM target was marginally superior to that of the C-ELM with the IRM target.

The performance of the C-ELM was also compared to the performance of a neural network trained with stochastic gradient descent (SGD). The SGD-trained network was better at utilizing the information provided by a large acoustic context window, particularly at low SNR levels, while a single frame of acoustic context was optimal for the C-ELM. In matched and mismatched noise tests, the C-ELM performed exceptionally well with a small amount of training data. The C-ELM with the IRM target outperformed all the other networks with just 1 hour

The C-ELM with the IRM target when trained with just 1 hour of data outperformed the other networks that were examined, and its performance was almost equivalent to that of the SGD-trained network that was trained with 10 hours of speech data.

CHAPTER 6

A NOISE PREDICTION AND TIME-DOMAIN SUBTRACTION APPROACH TO DEEP NEURAL NETWORK BASED SPEECH ENHANCEMENT

This chapter introduces the noise prediction and time-domain subtraction approach to speech enhancement using deep neural networks. The work presented in this chapter has been published in the *Proceedings of the 2017 International Conference On Machine Learning And Applications* [113].

6.1 Introduction

In this chapter, we begin a shift in focus from using the extreme learning machine to using deep neural networks for speech enhancement. As mentioned in Section 2.1, DNN-based speech enhancement systems can be grouped into three three broad paradigms, namely, spectral mapping, T-F masking, and multitask learning approaches. One salient feature of all the aforementioned approaches is that they utilize training targets based on the clean speech features. For low-SNR speech signals, however, such approaches are known to produce degraded estimates in weak energy speech segments where it is extremely challenging for the DNN to distinguish between speech and noise as the noisy speech is very similar to the pure noise [38].

The poor performance in low-SNR leads us to investigate a different approach, speech enhancement via noise prediction. This approach is motivated by two principal considerations: first and foremost is improving the low-SNR performance of the speech enhancement system. The rationale behind using a noise prediction approach to improve low-SNR performance is that at 0dB SNR, the noise power is about equal to the signal power on average. At still lower SNR values e.g. -5dB SNR or lower, it can be expected that several segments of the noisy speech signal to be enhanced are dominated by noise. Hence, it can be intu-

itively expected that at low SNR, it should be easier for the DNN to learn a mapping from the noisy signal spectra to the noise spectra and accurately predict the noise spectra. For higher-SNR signals, the noise is dominated by the speech signal, and noise prediction will not distort the speech signal.

The second consideration is the effect of using the noisy phase during the reconstruction of the enhanced speech signal. While the signal phase has been traditionally considered to be unimportant [114], recent studies have challenged that notion and have shown performance improvements from phase compensation and the use of oracle phase [115, 116]. For lower-SNR speech signals, the phase of the noisy speech signal can be expected to be dominated by the phase of the noise signal. Since we assume the noise is additive, the noisy signal can be expressed as [117]

$$\begin{aligned} Y(\omega) &= S(\omega) + D(\omega) \\ &= |S(\omega)|e^{j\phi_S(\omega)} + |D(\omega)|e^{j\phi_D(\omega)} \\ &= |Y(\omega)|e^{j\phi_Y(\omega)} \end{aligned} \tag{6.1}$$

where $Y(\omega)$, $X(\omega)$, and $D(\omega)$ are the complex spectral representations of the noisy speech, clean speech, and additive noise respectively, and the noisy phase,

$$\phi_Y = \tan^{-1} \frac{|S| \sin \phi_S + |D| \sin \phi_D}{|S| \cos \phi_S + |D| \cos \phi_D} \tag{6.2}$$

where the frequency domain variable has been suppressed to simplify the notation. Modifying (6.2), it is clear that

$$\phi_Y = \tan^{-1} \frac{|S|/|D| \sin \phi_S + \sin \phi_D}{|S|/|D| \cos \phi_S + \cos \phi_D} \approx \phi_D \tag{6.3}$$

when $|D| \gg |S|$. Consequently, we investigate the performance of a time-domain noise subtraction architecture in which the additive noise signal is reconstructed and the enhanced

speech waveform is obtained by time-domain subtraction of the predicted additive noise signal from the noisy speech signal.

The use of a noise prediction approach naturally raises a key question of how such a system might generalize, particularly to unseen noise types. We investigate this question by comparing the performance of the proposed noise prediction architectures to that of conventional spectral mapping architectures. The rest of the chapter is organized as follows: an overview of the proposed noise prediction systems is given in Section 6.2, experiments are described in Section 6.3, results are presented in Section 6.4, and discussions follow in Section 6.5. Conclusions are presented in Section 6.6.

6.2 System Overview

A block diagram of the proposed speech enhancement systems is shown in Figure 6.1. Three variants of a noise prediction framework are investigated. The baseline noise prediction system, which we term the time domain noise subtraction (TDS) system, is shown in Figure 6.1(a). In the training phase of the TDS system, log magnitude spectral features are extracted from the framed noisy speech and added noise signals. The noisy speech log spectra are fed into the neural network and the network learns a mapping function between the features of the noisy speech input and the spectral features of the added noise signal.

The network is trained by using the back-propagation algorithm to minimize a mean-square error criterion. Network parameters are updated using mini-batch stochastic gradient descent with momentum. The error criterion is

$$E = \frac{1}{N}\sum_{i=1}^{N} ||\hat{\mathbf{N}}_i(\mathbf{y}_i, \mathbf{\Theta}) - \mathbf{N}_i||^2, \tag{6.4}$$

where $\mathbf{y}_i$ is the input to the network, $\mathbf{\Theta} = \{\mathbf{W}, \mathbf{b}\}$, represents the weights and biases in the network, $\hat{\mathbf{N}}_i(\mathbf{y}_i, \mathbf{\Theta})$ is the output of the network, $\mathbf{N}_i$, the log magnitude spectrum of the corrupting noise, is the desired target, and N is the mini-batch size.

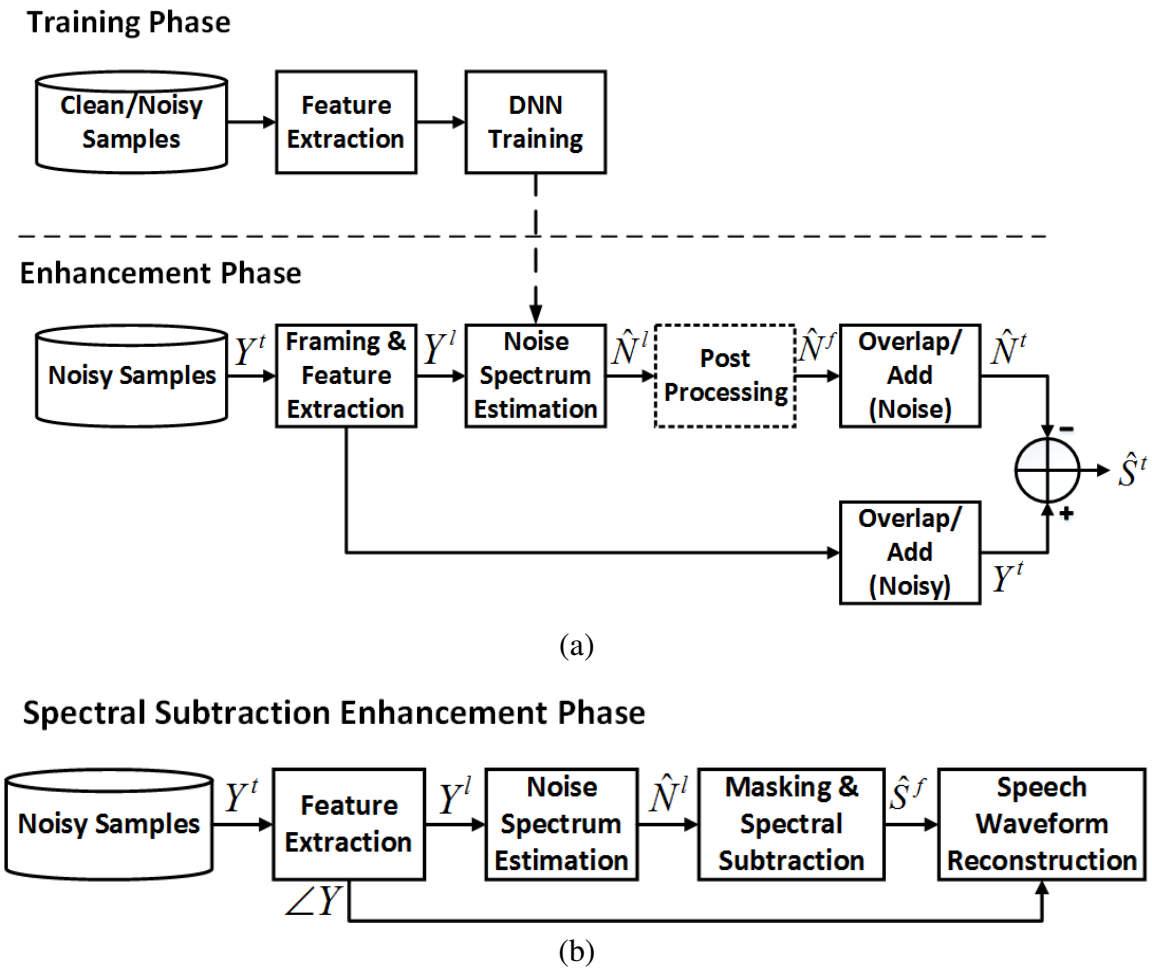


Figure 6.1: Block diagram of the proposed systems.

In the enhancement phase of the TDS system, log spectral features extracted from noisy speech frames are fed into the trained network, and the network predicts the log spectrum of the added noise in each frame. The predicted spectra are combined with the noisy phase and a time-domain additive noise signal estimate is synthesized using the overlap-add method [100]. A real-time system is implemented by using a separate overlap-add buffer for the synthesis of the noisy signal frames. The noise-free speech signal estimates are then obtained by subtracting the added noise estimate from the noisy speech signal as shown in Figure 1(a).

The first variant of the baseline system includes a mask-based post processing module and is also shown in Figure 6.1(a). The mask-based processing system (MBP) and the

baseline TDS system are identical in the training phase but differ in the enhancement phase. In the MBP enhancement phase, the network predicts the noise spectrum estimate for each frame in the same manner as is done in the baseline (TDS) system. The noise spectral estimates are then used to compute a time-frequency (T-F) mask which is computed as:

$$H(t,\omega) = \min\left\{\left(\frac{\hat{N}^2(t,\omega)}{Y^2(t,\omega)}\right)^{\frac{1}{2}}, 1\right\}, \tag{6.5}$$

where $\hat{N}^2(t,\omega)$ and $Y^2(t,\omega)$ represent the estimated noise and noisy speech signal power spectral densities respectively.

The new post-processed noise spectral estimates are then obtained as

$$\hat{N}_{pp}(t,\omega) = H(t,\omega)X(t,\omega), \tag{6.6}$$

where $X(t,\omega)$ is the noisy speech complex spectrum. The enhanced speech signal is then obtained using the same overlap-add method that was described for the TDS system. Some remarks about the post-processing mask in (6.5) are in order. The mask is computed by normalizing the estimated added noise signal power by the noisy signal power and enforcing an upper bound of unity. The mask thus represents a probability or confidence that a bin contains noise. The enforced upper bound also serves to prevent distortions that could be caused by estimation errors.

The second variant of the noise prediction speech enhancement framework, the spectral subtraction (SS) system, is shown in Figure 6.1(b). The SS system is also identical to the TDS system in the training phase, and the neural network learns a mapping function between the features of the noisy speech input and the features of the added noise signal. In the enhancement phase, the network predicts the added noise log spectra from the noisy speech log-spectral input features, the masking function, (6.5), is applied, and an estimate of the noise-free speech spectrum is obtained by spectral subtraction. Any negative spec-

tral values that result from the subtraction process are handled using a noise floor. The noise-free speech spectrum estimates are combined with the noisy phase, and the enhanced speech signal is then synthesized with the overlap-add method. The SS system thus allows us to assess the effect of reconstructing the enhanced speech signal instead of the added noise signal with the noisy phase and determine if there are any benefits from using the time-domain subtraction procedure.

6.3 Experiments

Table 6.1: Description of noise types used in testing.

Noise	Description	Noise	Description
n1	Crowd	n6	Water
n2	Machine	n7	Wind
n3	Alarm/Siren	n8	Bell
n4	Traffic/Car	n9	Cough
n5	Animal	n10	Clap

All experiments were performed using recorded sentences from the IEEE Corpus [98] included with the NOIZEUS database [8]. The corpus is comprised of 72 lists, each of which contains 10 sentences. Our noise samples came from a database of 100 non-speech sounds [99]. Both the noise-free speech and noise recordings were resampled to 8kHz. The training datasets were comprised of sentences taken from lists 1 - 60, while testing was done with the 50 sentences from lists 68 - 72.

Four training datasets were created by adding noise to the clean speech sentences. The first two datasets, made for the conventional spectral mapping models, were created by respectively adding 10 and 50 noise types to clean speech samples at six SNR levels ranging from 20dB to -5dB in 5dB steps. These training sets were about 25 hours and 50 hours in length respectively. Two similar datasets were created for training the noise prediction models. The training sets were identical in length, but were created by added the noise at

seven SNR levels ranging from 20dB to -10dB in 5dB steps instead. The additional SNR level was added to increase the number of training samples with a strong representation of the added noise signals.

The speech signals were divided into 32ms frames and spectral features extracted from the clean speech, noisy speech, and from the added noise signals were used to create noisy and noise-free speech, and noisy speech and added noise log spectral pairs. Fourier analysis was performed using a Hamming window. The proposed noise prediction models used log magnitude spectral features, while the conventional models used log power spectral features following the common practice. Input features were normalized to have zero mean and unit variance, while training targets were not normalized.

To allow the networks to take advantage of temporal information, each input vector included adjacent time frames. Consequently, each input vector was constructed as

$$\mathbf{y}_i = [\mathbf{x}_{i-l}, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_{i+l}]. \tag{6.7}$$

Five context frames, i.e. $l = 5$, for an input length of eleven frames, were used in training and evaluation of the enhancement systems.

The conventional models included a network trained with only with noisy log-power spectral inputs (LPS) and a noise-aware trained (NAT) model in which the input feature vector was expanded by appending an estimate of the noise in each utterance [6, 19]. The noise estimate, $\hat{\mathbf{n}}_i$, was fixed for each utterance and was obtained by averaging the first five frames of noisy speech log spectra as

$$\hat{\mathbf{n}}_i = \hat{\mathbf{n}} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{x}_k. \tag{6.8}$$

The neural network models were all deep neural networks with three hidden layers, each containing 2000 hidden nodes. The hidden layers of all the networks used the rectified linear unit (ReLU) activation functions [107, 108], and the output layers were linear.

Weights and biases of all the layers were initialized following the method of He *et al*. [109], and the networks were trained using gradient descent with momentum. The initial learning rate was set to 0.001 for the first 10 epochs, then decreased by 10% every subsequent 10 epochs, and the the momentum coefficient was set for 0.9. A mini-batch size of 128 samples was used, and the networks were trained for 50 epochs. All networks were implemented and trained using the TensorFlow library [110].

Testing was done using both seen and unseen noise types. Ten noise types were used in each of the testing scenarios. In the seen noise tests, each of the noise types used during the enhancement or evaluation phase was one of the noise types used during the training phase. Conversely, in unseen noise testing, each of the noise types used during the evaluation phase had not been used during the training of the network. A description of the noise types is given in Table 3.1.

Speech quality and intelligibility were objectively evaluated using the perceptual evaluation of speech quality (PESQ) [101] and short-time objective intelligibility (STOI) [118] metrics respectively. PESQ scores range from -0.5 to 4.5 while STOI scores range from 0 to 1. These measures have been shown to have high correlation with subjective listening tests [119, 120] .

6.4 Results

6.4.1 Evaluation in Seen Noise

The PESQ results for the different models are presented in Table 6.2. First examining the results of the noise prediction (NP) models, the PESQ scores of the MBP and TDS models show that the mask-based processing approach improves the quality of the enhanced speech at all SNR levels. The greatest improvement, however, is seen at mid-range SNR input levels. The PESQ scores also show that there is no perceptual difference between the MBP and SS models, hence, there was no advantage gained by reconstructing the added noise instead of the enhanced speech signal with the noisy phase.

Table 6.2: Average PESQ scores for the conventional and proposed systems trained with the 10-noise dataset in seen noise. The average over all SNR levels is denoted AVG. The LPS and NAT models are collectively referred to as the speech prediction (SP) models, and the TDS, MBP, and SS models are collectively referred to as the noise prediction (NP) models.

SNR (dB)	Noisy	LPS	NAT	TDS	MBP	SS
20	3.027	3.837	3.824	3.774	3.874	3.874
15	2.701	3.720	3.710	3.558	3.709	3.709
10	2.380	3.576	3.566	3.315	3.493	3.493
5	2.072	3.404	3.395	3.066	3.242	3.242
0	1.791	3.200	3.192	2.817	2.964	2.964
-5	1.503	2.960	2.949	2.553	2.692	2.692
AVG.	2.246	3.449	3.439	3.180	3.329	3.329

Comparing the NP and speech prediction (SP) models, we see that the NP models rival the performance of the SP models at the higher SNR levels, and even perform better at the highest SNR level of 20dB, however, their relative performance rapidly declines as the SNR reduces. The drop in performance begins at about 10dB SNR. The conventional models perform better on average with LPS model exhibiting the best performance on the 10-noise dataset.

The STOI results presented in Table 6.3 show that post processing does not have much effect on the intelligibility of the enhanced speech signal leading to the virtually identical performance of the TDS, MBP, and SS models. A comparison of all models shows that the NP models perform better than the SP models at 5dB SNR and greater. All the models have virtually identical performance at 0dB SNR, and the SP models perform slightly better at -5dB SNR. On average, the NP models perform better with the MBP and SS models having the best performance.

The results for the training set with 50 noise types are shown in Table 6.4. The PESQ and STOI scores of the NP models follow the same trends that were observed in Tables 6.2 and 6.3. A comparison of the PESQ scores of all the models shows a greater difference in

Table 6.3: Average STOI scores for the conventional and proposed systems trained with the 10-noise dataset in seen noise.

SNR (dB)	Noisy	LPS	NAT	TDS	MBP	SS
20	0.961	0.967	0.967	0.984	0.985	0.985
15	0.926	0.958	0.959	0.972	0.974	0.974
10	0.872	0.946	0.947	0.955	0.958	0.958
5	0.799	0.928	0.929	0.931	0.934	0.934
0	0.708	0.903	0.904	0.897	0.9000	0.900
-5	0.608	0.868	0.868	0.854	0.857	0.857
AVG.	0.812	0.928	0.929	0.932	0.934	0.934

Table 6.4: Average PESQ and STOI scores for the conventional and proposed systems trained with the 50-noise dataset in seen noise.

SNR	PESQ						STOI					
(dB)	Noisy	LPS	NAT	TDS	MBP	SS	Noisy	LPS	NAT	TDS	MBP	SS
20	3.027	3.416	3.506	3.605	3.705	3.705	0.961	0.931	0.937	0.980	0.981	0.981
15	2.701	3.313	3.394	3.349	3.488	3.488	0.926	0.923	0.928	0.965	0.968	0.968
10	2.380	3.193	3.265	3.079	3.234	3.234	0.872	0.911	0.916	0.941	0.947	0.947
5	2.072	3.052	3.114	2.812	2.961	2.961	0.799	0.893	0.897	0.907	0.916	0.916
0	1.791	2.878	2.932	2.540	2.679	2.679	0.708	0.867	0.872	0.860	0.872	0.872
-5	1.503	2.653	2.708	2.261	2.377	2.377	0.608	0.829	0.834	0.801	0.814	0.814
AVG.	2.246	3.084	3.153	2.941	3.074	3.074	0.812	0.892	0.897	0.909	0.916	0.916

performance at the extreme SNR levels with the MBP and SS models performing better at higher SNR levels, and the LPS and NAT models performing better at lower SNR levels. The average PESQ scores of all the models are closer than in Table 6.2, and the NAT model has the best average performance. There is also greater difference in the STOI scores of the models, with the NP models performing better at the higher SNR levels and on average over all SNR levels. These results suggest that the NP models are better at enhancing the intelligibility of speech in seen noise types.

Table 6.5: Average PESQ scores for the conventional and proposed systems trained with the 10-noise dataset in unseen noise.

SNR (dB)	Noisy	LPS	NAT	TDS	MBP	SS
20	3.182	3.425	3.362	3.214	3.237	3.237
15	2.875	3.153	3.110	2.914	2.944	2.944
10	2.569	2.876	2.845	2.610	2.645	2.645
5	2.288	2.593	2.571	2.318	2.351	2.351
0	2.036	2.312	2.300	2.050	2.079	2.079
-5	1.779	2.031	2.031	1.796	1.817	1.817
AVG.	2.455	2.732	2.703	2.484	2.512	2.512

Table 6.6: Average STOI scores for the conventional and proposed systems trained with the 10-noise dataset in unseen noise.

SNR (dB)	Noisy	LPS	NAT	TDS	MBP	SS
20	0.958	0.958	0.957	0.956	0.959	0.959
15	0.925	0.940	0.938	0.924	0.928	0.928
10	0.876	0.908	0.904	0.875	0.880	0.880
5	0.813	0.858	0.854	0.808	0.816	0.816
0	0.736	0.789	0.786	0.727	0.737	0.737
-5	0.650	0.703	0.701	0.636	0.648	0.648
AVG.	0.826	0.859	0.857	0.821	0.828	0.828

6.4.2 Evaluation in Unseen Noise

The PESQ results in unseen noise for the dataset with 10 noise types are presented in Table 6.5. A comparison of the NP models shows that the PESQ scores of the NP models, unlike with the seen noise tests, are much closer, and there not as much benefit from post processing. The SP models perform better at all SNR levels and can be seen to have a clear performance advantage in unseen noise.

The STOI results in unseen noise for the dataset with ten noise types are presented in Table 6.6. Once again, all the NP models have virtually identical performance. The NP

Table 6.7: Average PESQ scores for the conventional and proposed systems trained with the 50-noise dataset in unseen noise.

SNR	PESQ						STOI					
(dB)	Noisy	LPS	NAT	TDS	MBP	SS	Noisy	LPS	NAT	TDS	MBP	SS
20	3.182	3.376	3.426	3.252	3.275	3.275	0.958	0.930	0.935	0.964	0.965	0.965
15	2.875	3.216	3.242	2.953	2.987	2.987	0.925	0.920	0.922	0.934	0.935	0.935
10	2.569	3.013	3.020	2.652	2.690	2.690	0.876	0.900	0.900	0.888	0.890	0.890
5	2.288	2.763	2.760	2.357	2.393	2.393	0.813	0.864	0.862	0.824	0.828	0.828
0	2.036	2.473	2.475	2.082	2.111	2.111	0.736	0.806	0.804	0.745	0.750	0.750
-5	1.779	2.173	2.182	1.813	1.837	1.837	0.650	0.726	0.727	0.654	0.662	0.662
AVG.	2.455	2.836	2.851	2.518	2.549	2.549	0.826	0.858	0.858	0.835	0.838	0.838

and SP models have similar performance at 20dB SNR, but the SP models show superior performance as SNR levels reduce.

The results for the training set with 50 noise types are presented in Table 6.7. The results of the NP models show the same trends in performance that were observed in Tables 6.5 and 6.6. The PESQ scores show that the SP models perform better than the NP models in enhancing the quality of speech at all SNR levels. Although the SP models are superior in enhancing the quality in unseen noise, the NP models, interestingly, show superior intelligibility performance at higher SNR levels. All models have virtually identical STOI scores at 10dB SNR, and the SP models outperform the NP models at lower SNR and are on average slightly better than the NP models.

6.5 Discussion

The NP models were proposed to improve on the low-SNR performance of neural network-based speech enhancement systems, however, when compared to the conventional SP models, they produced high quality speech at high-SNR levels, and performed poorly at low-SNR levels. They also performed well on intelligibility test metrics even in unseen noise conditions where they performed poorly on quality metrics. The reasons for these observations are not yet entirely clear and are being actively researched, however, we can offer

some insights.

The SP models predict the speech spectrum, and estimation errors could result in either attenuation or amplification distortions of the speech spectrum. Furthermore, severe amplification distortions of the speech signal spectrum have been shown to adversely affect speech intelligibility [10]. The NP models, on the other hand, predict the noise spectrum, and estimation errors could either result in the loss of some of the speech spectral components or in the presence of residual noise in the enhanced speech signal. The spectrograms in Figure 6.2 show that high SNR, the MBP model successfully removes the added noise, and appears to remove more of the added noise than the NAT model. The spectrogram of the MBP model in Figure 6.3, on the other hand, shows some areas where there is some loss of formant structure and others with more residual noise than can be seen in the NAT spectrogram in the same figure. Our informal listening tests confirmed that the lower-SNR speech samples, such as those at 0dB or -5dB average SNR, enhanced by the NAT model had a tendency to be garbled, while the speech produced by the MBP model was not garbled but had more residual noise. The NP models thus have a tendency to under-estimate the added noise for lower-SNR speech samples, and this can be explained by the fact that there is greater variation in the training targets of these models since the training target, i.e the added noise, is SNR-dependent. The SP models, on the hand, have a fixed target (the clean speech spectrum) that is not SNR-dependent.

The difference in training targets is also very likely responsible for the poor performance of the NP models relative to the SP models in unseen noise. The NP models are likely to be more susceptible to prediction errors when the unseen noise tested differs markedly from any of the noise types in the training set and means to overcome this limitation would be further investigated.

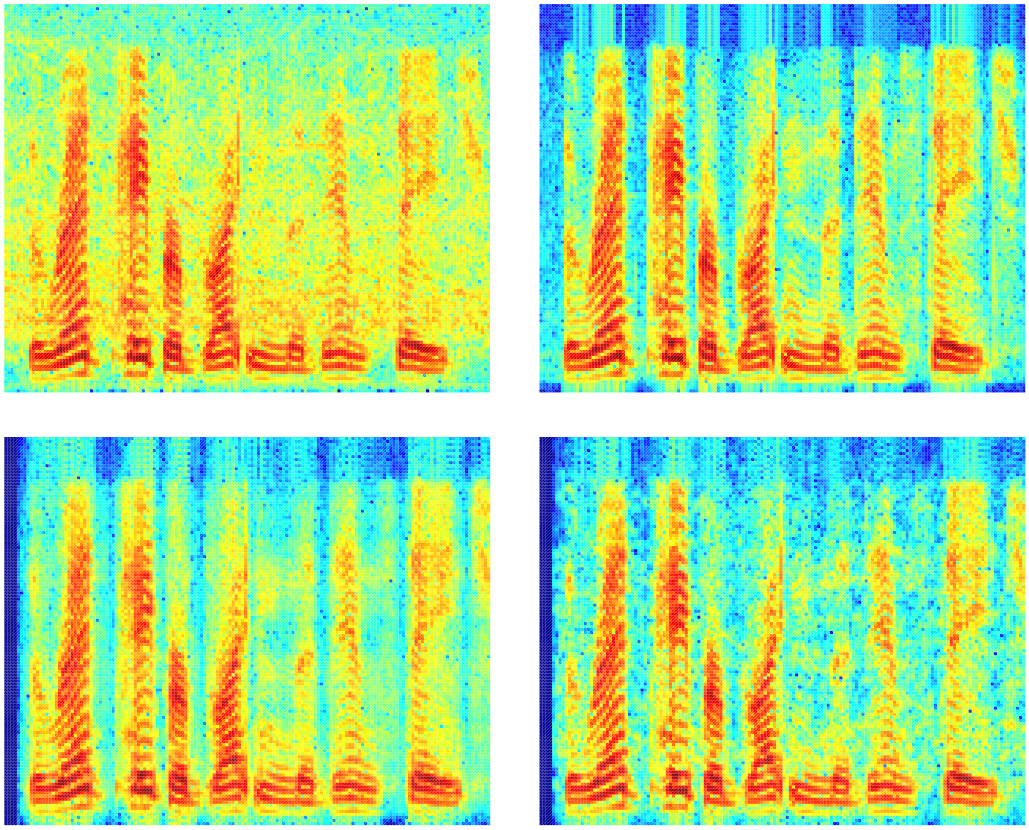


Figure 6.2: Example spectrograms of an utterance in seen crowd noise at 20dB. From upper left clockwise, noisy signal, clean signal, MBP enhanced, and NAT enhanced.

6.6 Conclusion

In this chapter we proposed an approach to speech enhancement based on noise prediction. A baseline system, the time-domain noise subtraction system, and two variations of the baseline system were implemented and compared to conventional spectral mapping models. The proposed systems outperformed the conventional systems on speech quality metrics at high-SNR levels in seen noise tests, but under-performed them at low-SNR levels and in unseen noise tests. They also exhibited strong performance on intelligibility metrics at high-SNR levels in both seen and unseen noise tests. The results show that there is a need to better understand and improve the performance of the noise prediction models. In particular, the use of different training targets and additional features for improving the

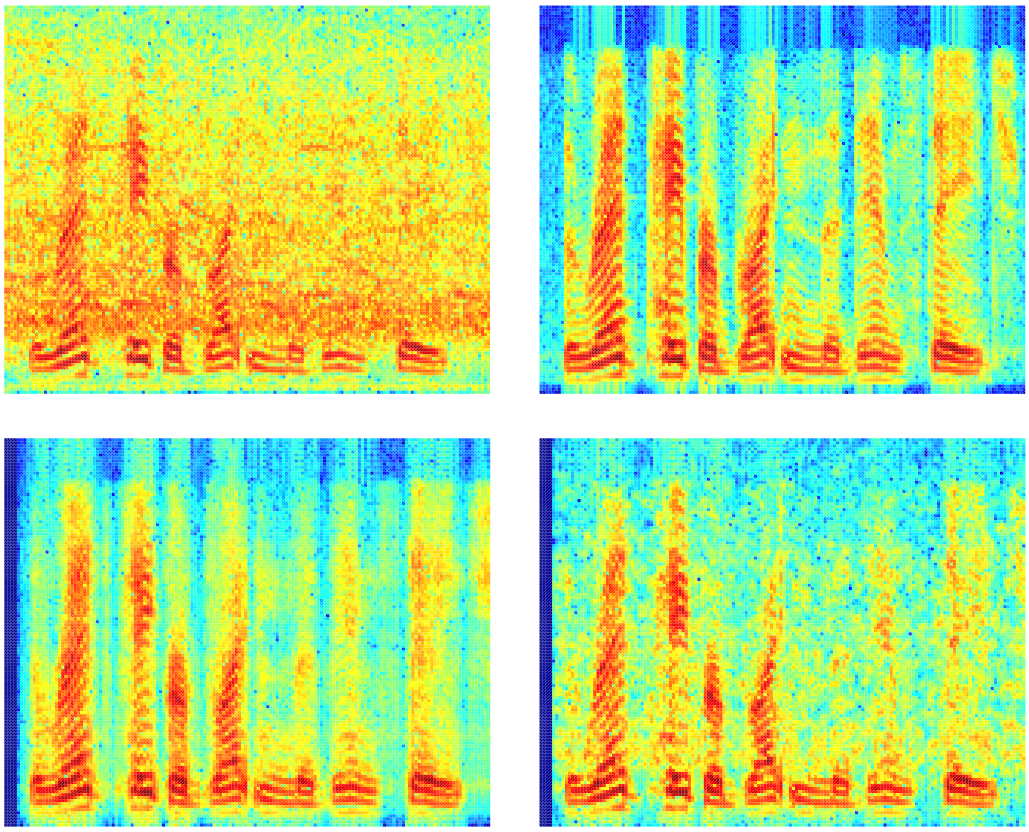


Figure 6.3: Example spectrograms of an utterance in seen crowd noise at 5dB. From upper left clockwise, noisy signal, clean signal, MBP enhanced, and NAT enhanced.

robustness of the noise prediction models in both low-SNR environments and in unseen noise conditions need to be investigated.

CHAPTER 7

A STUDY OF TRAINING TARGETS FOR DEEP NEURAL NETWORK-BASED SPEECH ENHANCEMENT USING NOISE PREDICTION

This chapter presents a study of different neural network training targets for the noise prediction and time-domain subtraction approach to speech enhancement introduced in the previous chapter. The work presented in this chapter has been published in the *2018 IEEE International Conference on Acoustics, Speech and Signal Processing* [121].

7.1 Introduction

In the previous chapter, we proposed noise-prediction and time-domain subtraction framework as an alternate approach to DNN-based speech enhancement. The rationale behind the use of the noise prediction approach was that learning a mapping between noisy speech input and added noise target features should be easier than learning a mapping between noisy speech input and the clean speech target features when the noise dominates the speech signal. The unexpected and somewhat contradictory performance of this approach, exhibiting stronger performance enhancing high-SNR signals than enhancing low-SNR signals, as well as the poor performance in unseen noise, indicated that the use of more robust features would be beneficial.

In this chapter, we evaluate the performance of different training targets for DNN speech enhancement based on noise prediction. Three training targets are examined, and their performance is compared to that of a DNN trained with a conventional clean speech target. The spectral mapping framework commonly referred to as noise-aware training (NAT) [6, 19] is used for this comparison. The rest of the paper is organized as follows: an overview of the proposed noise prediction systems is given in Section 7.2, experiments are described in Section 7.3, results are presented in Section 7.4, and conclusions are presented

in Section 7.5.

7.2 System Overview

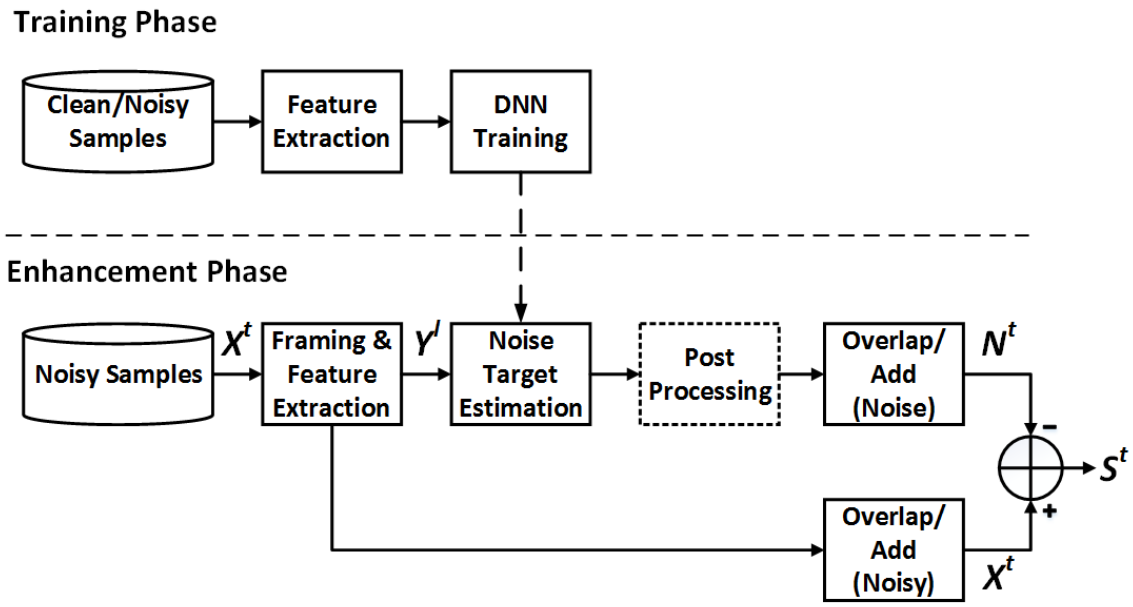


Figure 7.1: Block diagram of the proposed systems.

A block diagram of the proposed speech enhancement systems is shown in Figure 7.1. In the training phase, input-output feature pairs are extracted from the framed noisy speech and added noise signals respectively. Log magnitude spectral features are used as input features. Three training targets, namely, log spectral magnitude (LogFFT), Fourier magnitude spectrum mask (FFT-MASK), and a target which we introduce, the noise ratio mask (NRM), are evaluated.

1. Log Magnitude Spectrum

The magnitude of the short-time Fourier transform (STFT) spectrum of the noise is the natural choice for a training target in order to reconstruct the added noise. The STFT magnitude spectrum has a wide dynamic range, hence it is log compressed to reduce dynamic range and ease the DNN training process.

2. Fourier Magnitude Spectrum Mask

In conventional (speech prediction) spectral mapping models, the magnitude of the log spectral target is independent of the SNR of the noisy input signal since the target is the clean speech spectrum. The log spectral noise target, however, varies with SNR since the energy of the added noise depends on the SNR of the noisy input signal. The variation in the training target can be reduced by normalizing the magnitude spectrum of the added noise with that of the noisy speech signal. This gives the magnitude spectrum mask which is defined as:

$$M_{FFT}(t,\omega) = \frac{N(t,\omega)}{X(t,\omega)}, \tag{7.1}$$

where $M_{FFT}(t,\omega)$ is the mask, and $N(t,\omega)$ and $X(t,\omega)$ are the spectral magnitudes of the added-noise and noisy speech signals respectively. FFT-MASK is unbounded above, hence we enforced an upper bound to allow for more consistent training of the DNN. An upper bound of 3 was chosen by examining the distribution of a large random sample of the frequency bins.

3. Noise Ratio Mask

The noise ratio mask is defined as:

$$NRM(t,\omega) = \left(\frac{N^2(t,\omega)}{S^2(t,\omega) + N^2(t,\omega)}\right)^{\frac{1}{2}}, \tag{7.2}$$

where $N^2(t,\omega)$ and $S^2(t,\omega)$ represent the added-noise and speech signal power spectral densities respectively. The NRM is a bounded target with the range of [0,1], and can be seen to be equivalent to the frequency domain square-root Wiener filter if the speech and additive noise are assumed to be uncorrelated, and the noise is considered as the desired signal.

The network is trained by using the back-propagation algorithm to minimize a mean-square error criterion. Network parameters are updated using mini-batch stochastic gradi-

ent descent with momentum. The error criterion is

$$E = \frac{1}{N}\sum_{i=1}^{N} ||\hat{\mathbf{T}}_i(\mathbf{y}_i, \mathbf{\Theta}) - \mathbf{T}_i||^2 + \frac{\lambda}{2}||\mathbf{W}||_2^2 \tag{7.3}$$

where $\mathbf{y}_i$ is the input to the network, $\mathbf{\Theta} = \{\mathbf{W}, \mathbf{b}\}$, represents the weights and biases in the network, $\hat{\mathbf{T}}_i(\mathbf{y}_i, \mathbf{\Theta})$ is the output of the network, $\mathbf{T}_i$ is the desired training target, λ is the regularization coefficient, and N is the mini-batch size.

In the enhancement phase, log spectral features extracted from noisy speech frames are fed into the trained network, and the network computes an estimate of the desired training target vector. For the log spectral target, LogFFT, a post-processing step follows [113, 122]. The magnitude spectrum estimates are used to compute a time-frequency (T-F) mask which is computed as:

$$H(t,\omega) = \min\left\{\left(\frac{\hat{N}^2(t,\omega)}{X^2(t,\omega)}\right)^{\frac{1}{2}}, 1\right\}, \tag{7.4}$$

where $\hat{N}^2(t,\omega)$ and $X^2(t,\omega)$ represent the estimated noise and noisy speech signal power spectral densities respectively. The mask, (7.4), is computed by normalizing the estimated added-noise signal power by the noisy signal power and enforcing an upper bound of unity. The mask thus represents a probability or confidence that a bin contains noise. The enforced upper bound also serves to prevent distortions that could be caused by estimation errors. The new post-processed noise spectral estimates are then obtained as:

$$\hat{N}_{pp}(t,\omega) = H(t,\omega)X(t,\omega), \tag{7.5}$$

where $X(t,\omega)$ is the noisy speech magnitude spectrum.

For the FFT-MASK and NRM targets, the noise spectral estimates are obtained by

multiplying the predicted mask by the magnitude spectrum of the noisy speech as:

$$\hat{N}(t,\omega) = \hat{M}_{TF}(t,\omega)X(t,\omega) \tag{7.6}$$

where M_{TF} represents either the FFT-MASK or NRM targets. The predicted spectra are combined with the noisy phase, and a time-domain additive noise signal estimate is synthesized using the overlap-add method [100]. A real-time system is implemented by using a separate overlap-add buffer for the synthesis of the noisy signal frames. The noise-free speech signal estimates are then obtained by subtracting the added-noise signal estimate from the noisy speech signal as shown in Figure 7.1.

7.3 Experiments

Table 7.1: Description of noise types used in testing.

Noise	Description	Noise	Description
n1	Crowd	n6	Water
n2	Machine	n7	Wind
n3	Alarm/Siren	n8	Bell
n4	Traffic/Car	n9	Cough
n5	Animal	n10	Clap

All experiments were performed using recorded sentences from the IEEE Corpus [98] included with the NOIZEUS database [8]. The corpus is comprised of 72 lists, each of which contains 10 sentences. Our noise samples came from a database of 100 non-speech sounds [99]. Both the noise-free speech and noise recordings were resampled to 8kHz. The training datasets were comprised of sentences taken from lists 1 - 60, while testing was done with the 50 sentences from lists 68 - 72.

Four training datasets were created by adding noise to the clean speech sentences. The first three datasets, which include those made for the NAT, FFT-MASK, and NRM models,

were created by adding 50 noise types to the chosen clean speech samples at six SNR levels ranging from 20dB to -5dB in 5dB steps. The length of each dataset was about 50 hours. A similar-length training dataset was also created for the log magnitude spectral target. This dataset was, however, created by added the noise at seven SNR levels ranging from 20dB to -10dB in 5dB steps instead. The additional SNR level was added to increase the number of training samples that had a strong representation of the added-noise signals.

The speech signals were divided into 32ms frames and spectral features extracted from the clean speech, noisy speech, and from the added-noise signals were used to create input-output pairs for training the networks. Fourier analysis was performed using a Hamming window. The proposed noise prediction models used log magnitude spectral input features and targets as described in section 7.2, while the NAT model used log power spectral features following the common practice.

To allow the networks to take advantage of temporal information, each input vector included adjacent time frames. Consequently, each input vector was constructed as

$$\mathbf{y}_i = [\mathbf{x}_{i-l}, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_{i+l}]. \tag{7.7}$$

Five context frames, i.e. $l = 5$, for a total input length of eleven frames, were used in the training and evaluation of the enhancement systems.

The spectral input vectors for the NAT model were created by appending an estimate of the noise in each utterance to the noisy signal spectral input (7.7). The noise estimate, $\hat{\mathbf{n}}_i$, was fixed for each utterance and was obtained by averaging the first five frames of noisy speech log spectra as

$$\hat{\mathbf{n}}_i = \hat{\mathbf{n}} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{x}_k. \tag{7.8}$$

The neural network models were all deep networks with three hidden layers, each containing 2000 hidden nodes. The hidden layers of all the networks used the rectified linear unit (ReLU) activation functions [107, 108], and the output layers were linear. Weights

and biases of all the layers were initialized following the method of He *et al*. [109], and the networks were trained using gradient descent with momentum. The initial learning rate was set to 0.001 for the first 10 epochs, and then decreased by 10% every subsequent 10 epochs. The value of the regularization coefficient was set to 0.0001, and the the momentum coefficient was 0.9. A mini-batch size of 128 samples was used, and the networks were trained for 50 epochs. All networks were implemented and trained using the TensorFlow library [110].

Testing was done using both seen and unseen noise types. Ten noise types were used in each of the testing scenarios. In the seen noise tests, each of the noise types used during the enhancement or evaluation phase was one of the noise types used during the training phase. Conversely, in unseen noise testing, each of the noise types used during the evaluation phase had not been used during the training of the network. A description of the noise types is given in Table 7.1.

Speech quality and intelligibility were once again objectively evaluated using the perceptual evaluation of speech quality (PESQ) [101] and short-time objective intelligibility (STOI) [118] metrics respectively. PESQ scores range from -0.5 to 4.5 while STOI scores range from 0 to 1. These measures have been shown to have high correlation with subjective listening tests [119, 120].

7.4 Results

7.4.1 Evaluation in Seen Noise

The average PESQ scores for all the models in seen noise conditions are shown in Table 7.2. The LogFFT and FFT-MASK models are similar in performance, but FFT-MASK has a slight edge when average SNR is above 5dB, and LogFFT has a slight advantage otherwise. The overall average scores for both methods are basically equivalent. The NRM performs better than both LogFFT and FFT-MASK at all SNR levels and is the best of the noise prediction models in enhancing speech quality.

The average STOI scores for all the models in seen noise conditions are shown in Table 7.3. LogFFT performs better than FFT-MASK at all input SNR levels and has a average STOI score that is about 1.5% higher. The difference in performance between these two training targets also increases as SNR reduces. The LogFFT model also performs slightly better than the NRM model, however, with an average STOI difference that is always less than 1% , the difference can be seen to be insignificant. Considering both the PESQ and STOI scores, the NRM model performs best in seen noise conditions, followed by the LogFFT and then the FFT-MASK models.

Table 7.2: Average PESQ scores for the different training targets and the noise aware training (NAT) models in seen noise conditions. The average over all SNR levels is denoted AVG.

SNR (dB)	Noisy	NAT	LogFFT	FFT-MASK	NRM
20	3.027	3.506	3.686	3.720	**3.765**
15	2.701	3.394	3.481	3.511	**3.590**
10	2.380	3.265	3.240	3.258	**3.380**
5	2.072	3.114	2.982	2.975	**3.134**
0	1.791	**2.932**	2.708	2.665	2.845
-5	1.503	**2.708**	2.409	2.327	2.513
AVG.	2.246	3.153	3.084	3.076	**3.205**

Table 7.3: Average STOI scores for the proposed and NAT systems in seen noise conditions.

SNR (dB)	Noisy	NAT	LogFFT	FFT-MASK	NRM
20	0.961	0.937	**0.981**	0.974	0.977
15	0.926	0.928	**0.968**	0.958	0.962
10	0.872	0.916	**0.947**	0.934	0.941
5	0.799	0.897	**0.917**	0.899	0.910
0	0.708	0.872	**0.874**	0.851	0.868
-5	0.608	**0.834**	0.817	0.787	0.808
AVG.	0.812	0.897	**0.917**	0.901	0.911

7.4.2 Evaluation in Unseen Noise

The average PESQ scores for all the models in unseen noise conditions are presented in Table 7.4. Unlike in seen noise conditions, FFT-MASK performs better than LogFFT at all SNR values and is consequently better on average. The NRM model is once again better than both the FFT-MASK and LogFFT models and is the best of the noise prediction models in enhancing speech quality.

The average STOI scores for all the models in unseen noise are shown in Table 7.5. FFT-MASK performs slightly better than LogFFT, but the average STOI difference is insignificant. The NRM model outperforms both the FFT-MASK and LogFFT models with a difference of about 2% in the average STOI scores.

Considering the performance of all the noise prediction models in both seen and noise noise conditions, the NRM model preforms best, followed by the FFT-MASK, and lastly, the LogFFT models. The two normalized models, NRM and FFT-MASK, perform markedly better than LogFFT in unseen noise conditions. This could be because their training targets are related to both the added-noise and the noisy signal spectra, and the noisy signal spectrum, in effect, constrains the value of the target. In unseen noise conditions, the constraining effect remains and the targets generalize better. This is not the case with LogFFT, and it is therefore more susceptible to prediction errors in unseen noise conditions.

7.4.3 Comparison of Speech and Noise Prediction Models

The PESQ scores in Table 7.2 show that the noise-prediction models perform comparatively well at higher SNR values, and in seen noise conditions. The NRM model outperforms the NAT model at all SNR values above 0dB, however, the NAT model performs better at the lower SNR values. The difference between the PESQ scores of the NAT and noise models as SNR decreases is worse for the LogFFT and FFT-MASK models than it is for the NRM model. The likely reason for the observed drop in performance with SNR is that the training targets of the noise prediction models are SNR dependent. As such, the DNN might tend

Table 7.4: Average PESQ scores for the proposed and NAT systems in unseen noise conditions.

SNR (dB)	Noisy	NAT	LogFFT	FFT-MASK	NRM
20	3.182	3.426	3.292	3.413	**3.530**
15	2.875	3.242	3.007	3.134	**3.266**
10	2.569	**3.020**	2.715	2.842	2.976
5	2.288	**2.760**	2.420	2.538	2.664
0	2.036	**2.475**	2.137	2.233	2.345
-5	1.779	**2.182**	1.865	1.942	2.032
AVG.	2.455	**2.851**	2.573	2.684	2.802

Table 7.5: Average STOI scores for the proposed and NAT systems in unseen noise conditions.

SNR (dB)	Noisy	NAT	LogFFT	FFT-MASK	NRM
20	0.958	0.935	0.965	0.965	**0.970**
15	0.925	0.922	0.937	0.939	**0.949**
10	0.876	0.900	0.893	0.899	**0.915**
5	0.813	0.862	0.832	0.842	**0.863**
0	0.736	**0.804**	0.755	0.767	0.793
-5	0.650	**0.727**	0.666	0.677	0.706
AVG.	0.826	0.858	0.841	0.848	**0.866**

to average over these targets leading to under-estimation of the noise in low-SNR signals. Our informal listening tests confirmed that the low-SNR speech signals enhanced by the noise prediction models had more residual noise than those enhanced by the NAT model.

The STOI scores in Table 7.3 show that that the noise prediction models also perform well in enhancing intelligibility in seen noise conditions. The LogFFT model outperforms the NAT model at all SNR values except -5dB, and both the LogFFT and NRM models outperform the NAT model by about 2% on average.

The PESQ scores in Table 7.4 show that the NRM model performs slightly better than the NAT model above 10dB SNR in unseen noise conditions. 10dB SNR marks an in-

flection point at which the NAT model becomes increasingly better than the NRM model as SNR decreases, and the NAT model performs slightly better than the NRM model on average. The STOI scores in Table 7.5 show the NRM model performs better than the NAT model above 0dB SNR, but the performance margin reduces as SNR decreases. The average STOI score of the NRM model is about 1% better than that of the NAT model.

The noise models can thus be seen to perform comparatively well to the NAT model in enhancing speech quality at higher average SNR values and in enhancing intelligibility even when the latter is not accompanied by corresponding quality enhancement. The most likely reason for this observation lies in how target estimation errors differently affect both model types. Estimation errors in the NAT model could either attenuate or amplify portions of the speech signal spectrum and cause attending distortions in the enhanced speech. Amplification distortions of the enhanced speech spectrum have been shown to adversely affect the speech intelligibility [10]. Estimation errors could similarly affect the estimated noise spectrum, however, these are more likely to occur in noise-dominant speech segments and do not affect the enhanced speech spectrum. Our informal listening tests showed that while the lower-SNR signal enhanced by the NAT model tended to be garbled, this was not the case with the noise models.

7.5 Conclusion

A study of DNN training targets for noise prediction was conducted. Objective test results showed the noise models were particularly effective in enhancing the intelligibility of noisy speech signals. The mask-based noise targets, which inherently include a normalization factor, performed better than the spectral noise target in unseen noise conditions. The noise ratio mask was the best all-round noise target. It outperformed the NAT model in seen noise conditions and in improving intelligibility in unseen noise, but fell short at lower SNR values.

CHAPTER 8
IMPROVING THE ROBUSTNESS OF NOISE PREDICTION MODELS WITH NOISE-AWARE TRAINING

8.1 Introduction

In this chapter we investigate the use of noise-aware training as a technique for increasing the robustness of the noise predictions models that have been developed in the previous two chapters. Noise-aware training (NAT) is a method of incorporating environmental information into the DNN-training process. It is done by augmenting the training features with an estimate of the noise in the utterance from which the features are derived. The method was introduced in the speech recognition literature [6], but it has also been applied in DNN-based speech enhancement [19, 123].

Two NAT methods, namely static and dynamic NAT, have been proposed for speech enhancement. Static NAT uses a fixed estimate of the noise, typically obtained from the first speech-free frames of an utterance, for all the speech frames obtained from the utterance. It is a direct adaptation of the NAT method as it was used in the training of a DNN-based ASR system, with the end goal of making the system more noise robust [6]. Static NAT, however, relies on the assumption that the noise is stationary over the duration of the utterance. While this may be a reasonable assumption in speech recognition where the utterances are short, single frames, of the order of milliseconds, representing a distinct sound or phoneme, it is not the case in speech enhancement where the goal is to suppress noise over long periods of several seconds or even minutes. Dynamic NAT, on the other hand, uses an estimate of the noise that is specific to each frame of speech obtained from an utterance. It is, therefore, more compatible with speech enhancement, however, it requires a means of estimating the noise present in each frame. In prior work by Y. Xu *et al.* [123], noise estimates were

obtained using an MMSE-based noise estimation algorithm or by computing a mask from the output of a DNN. The published results showed that the performance of the MMSE-based dynamic NAT system was inferior to the static NAT system, and the DNN-based system was superior to the static NAT system. Although the DNN-based system proved to be superior, the computation of the mask involved manually determining a threshold that defined speech and noise regions. In contrast, the noise ratio mask, defined in the previous chapter, learns such a T-F mask from the data and can be readily applied to dynamic noise estimation. Consequently, we investigate using a noise prediction network for dynamic noise estimation.

The noise-prediction architecture introduced in Chapter 6 performed well at high SNR levels but performed poorly at low SNR levels. It was conjectured that a reason for this observation is that the noise prediction models, unlike speech prediction models, have an SNR-dependent training target, and this causes an under-estimation of the noise present in the utterance. The model also performed poorly in unseen noise conditions. Although the performance of the noise prediction architecture was markedly improved by the use of more suitable training targets in the last chapter, a comparison of the noise and speech prediction models showed that there was still room for further improvement both in low SNR and unseen noise conditions. There are thus two driving factors behind the use of noise-aware training: firstly, the appended noise estimate could provide information on the level of noise present in the utterance and thus prevent under-estimation. Secondly, the noise estimate could act as a type of "noise signature" that would be useful in unseen noise conditions. The rest of the chapter is organized as follows: in the next section, we describe the static and dynamic NAT approaches that were investigated. Experiments are described in Section 8.3, results are presented in Section 8.4, and conclusions are presented in Section 8.5.

8.2 System Overview

8.2.1 Static Noise-aware Training

Static noise-aware training (NAT) has been previously described in Section 7.3 in the process of describing the speech prediction model, i.e., the NAT model that was used for performance comparisons. The static noise-aware noise prediction models follow the same paradigm, hence the noise estimate is obtained as

$$\hat{\mathbf{n}}_i = \hat{\mathbf{n}} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{x}_k \tag{8.1}$$

where $\mathbf{x}_k$ represents a speech frame. The first five frames, i.e., $K = 5$, of each utterance were averaged to produce the noise estimate. The networks were once again allowed to take advantage of temporal information by including adjacent time frames. Consequently, each input vector was constructed as

$$\mathbf{y}_i = [\mathbf{x}_{i-l}, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_{i+l}, \hat{\mathbf{n}}_i]. \tag{8.2}$$

8.2.2 Dynamic Noise-aware Training

A block diagram of the NAT noise prediction system is shown in Figure 8.1. During the training phase, each DNN input is augmented with an oracle noise estimate computed from the features. Two variations of the system were investigated: in the first variation, the noise ratio mask (NRM), which was defined in Section 7.2 as

$$NRM(t, \omega) = \left(\frac{N^2(t, \omega)}{S^2(t, \omega) + N^2(t, \omega)} \right)^{\frac{1}{2}}, \tag{8.3}$$

was computed for the current frame and used to augment the noisy speech (log magnitude spectral) features. In the second variation of the system, the log magnitude spectrum of the added noise in the current frame is used to augment the noisy features.

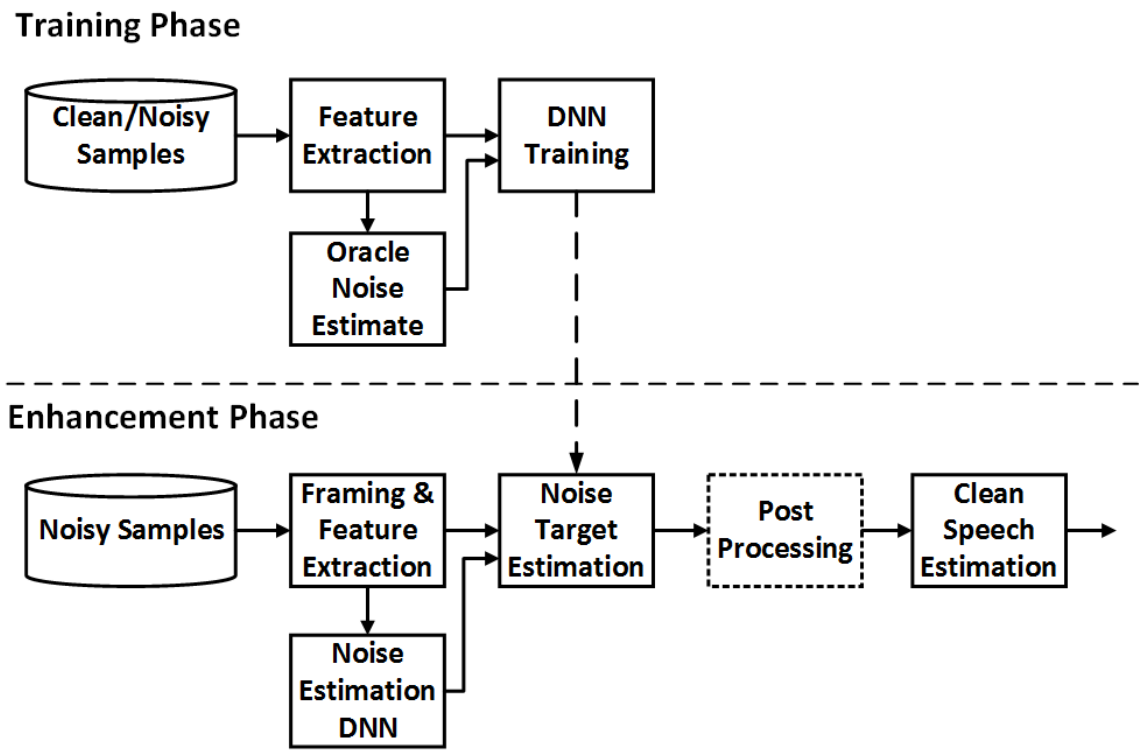


Figure 8.1: Block diagram of the NAT noise prediction system.

The network is trained by using the back-propagation algorithm to minimize a mean-square error criterion. Network parameters are updated using mini-batch stochastic gradient descent with momentum. The error criterion is

$$E = \frac{1}{N}\sum_{i=1}^{N} ||\hat{\mathbf{T}}_i(\mathbf{y}_i, \mathbf{\Theta}) - \mathbf{T}_i||^2 + \frac{\lambda}{2}||\mathbf{W}||_2^2 \tag{8.4}$$

where $\mathbf{y}_i$ is the input to the network, $\mathbf{\Theta} = \{\mathbf{W}, \mathbf{b}\}$, represents the weights and biases in the network, $\hat{\mathbf{T}}_i(\mathbf{y}_i, \mathbf{\Theta})$ is the output of the network, $\mathbf{T}_i$ is the desired training target, λ is the regularization coefficient, and N is the mini-batch size.

In the enhancement phase, the oracle noise information is no longer available and the appended noise features must, therefore, be estimated. Another DNN, a noise estimation DNN provides these estimates. A well-trained noise DNN with a noise ratio mask (NRM) target was used as the noise estimation DNN. For the first variation of the dynamic NAT

system, the output of the noise estimation DNN, which is an estimate of the value of the noise ratio mask, was appended to the noisy speech input features. The composite vector then became the input to the second DNN, the noise target estimation DNN. For the second variation of the dynamic NAT system, the noise estimation DNN was used to compute an estimate of the added noise magnitude spectrum. The noise spectral features were computed as

$$\widehat{N}(t,\omega) = \widehat{NRM}_{TF}(t,\omega)X(t,\omega) \tag{8.5}$$

where $\widehat{NRM}_{TF}(t,\omega)$ represent the output of the noise estimation DNN, and $X(t,\omega)$ represents the magnitude spectrum of the noisy speech. The output of the noise target estimation DNN is then used to enhance noisy speech spectra as was described in Section 7.2.

8.3 Experiments

Table 8.1: Description of noise types used in testing.

Noise	Description	Noise	Description
n1	Crowd	n6	Water
n2	Machine	n7	Wind
n3	Alarm/Siren	n8	Bell
n4	Traffic/Car	n9	Cough
n5	Animal	n10	Clap

All experiments were performed using recorded sentences from the IEEE Corpus [98] included with the NOIZEUS database [8]. The corpus is comprised of 72 lists, each of which contains 10 sentences. Our noise samples came from a database of 100 non-speech sounds [99]. Both the noise-free speech and noise recordings were resampled to 8kHz. The training datasets were comprised of sentences taken from lists 1 - 60, while testing was done with the 50 sentences from lists 68 - 72.

Training datasets with a length of about 100 hours were created by adding 50 noise types to the chosen clean speech samples at different SNR levels. The datasets for the

FFT-MASK, and NRM models were once again created using six SNR levels ranging from 20dB to -5dB in 5dB steps, while the dataset for the log magnitude spectral target was created using seven SNR levels ranging from 20dB to -10dB in 5dB steps.

The speech signals were divided into 32ms frames with 50% overlap, and spectral features extracted from the clean speech, noisy speech, and from the added-noise signals were used to create input-output pairs for training the networks. Fourier analysis was performed using a Hamming window. It should be noted that while the network inputs were changed in order to facilitate noise-aware training, the training targets remained the same. A complete description of the noise targets was provided in Section 7.2.

All the neural network models were all deep networks with three hidden layers, each containing 2000 hidden nodes. The hidden layers of all the networks used the rectified linear unit (ReLU) activation functions [107, 108], and the output layers were linear. Weights and biases of all the layers were initialized following the method of He *et al.* [109], and the networks were trained using gradient descent with momentum. The initial learning rate was set to 0.001 for the first 10 epochs, and then decreased by 10% every subsequent 10 epochs. The value of the regularization coefficient was set to 0.0001, and the the momentum coefficient was 0.9. A mini-batch size of 128 samples was used, and the networks were trained for 50 epochs. All networks were implemented and trained using the TensorFlow library [110].

Testing was done using both seen and unseen noise types. Ten noise types were used in each of the testing scenarios. In the seen noise tests, each of the noise types used during the enhancement or evaluation phase was one of the noise types used during the training phase. Conversely, in unseen noise testing, each of the noise types used during the evaluation phase had not been used during the training of the network. A description of the noise types is given in Table 8.1.

Speech quality and intelligibility were once again objectively evaluated using the perceptual evaluation of speech quality (PESQ) [101] and short-time objective intelligibility

(STOI) [118] metrics respectively. PESQ scores range from -0.5 to 4.5 while STOI scores range from 0 to 1. These measures have been shown to have high correlation with subjective listening tests [119, 120] .

8.4 Results and Discussion

8.4.1 Static NAT Evaluation

Table 8.2: Average PESQ scores for the different training targets and the static noise-aware training (NAT) models in seen noise conditions. The average over all SNR levels is denoted AVG.

SNR (dB)	Noisy	LogFFT	NAT-LogFFT	FFT-MASK	NAT-FFT-MASK	NRM	NAT-NRM
20	2.994	3.707	3.748	3.755	3.788	3.818	3.860
15	2.668	3.511	3.567	3.549	3.592	3.650	3.688
10	2.347	3.275	3.338	3.295	3.346	3.446	3.474
5	2.043	3.016	3.076	3.007	3.066	3.201	3.212
0	1.766	2.734	2.789	2.690	2.758	2.912	2.898
-5	1.459	2.427	2.472	2.342	2.418	2.581	2.540
AVG.	2.213	3.112	3.116	3.106	3.161	3.268	3.279

Table 8.3: Average STOI scores for the different training targets and the static noise-aware training (NAT) models in seen noise conditions. The average over all SNR levels is denoted AVG.

SNR (dB)	Noisy	LogFFT	NAT-LogFFT	FFT-MASK	NAT-FFT-MASK	NRM	NAT-NRM
20	0.958	0.981	0.982	0.975	0.977	0.977	0.979
15	0.921	0.968	0.970	0.959	0.962	0.963	0.966
10	0.866	0.948	0.951	0.935	0.939	0.942	0.945
5	0.790	0.918	0.923	0.901	0.907	0.913	0.916
0	0.698	0.876	0.883	0.855	0.862	0.871	0.874
-5	0.597	0.818	0.828	0.791	0.801	0.812	0.815
AVG.	0.805	0.918	0.923	0.903	0.908	0.913	0.916

The PESQ results for the static NAT systems in seen noise are presented in Table 8.2. A small, but insignificant, increase in the PESQ scores at all SNR levels is obtained by the use of static NAT in both the LogFFT and FFT-MASK models. A similar increase can be seen at all SNR levels except 0dB and -5dB in the NRM model. On average, FFT-MASK saw the greatest improvement, while LogFFT saw the least improvement.

The STOI results for the static NAT systems in seen noise are presented in Table 8.3. There is also a small, but insignificant, increase in the the STOI scores at all SNR levels for all the models. On average, LogFFT and FFT-MASK had the most improvement, while NRM had the least improvement.

Table 8.4: Average PESQ scores for the different training targets and the static noise-aware training (NAT) models in unseen noise conditions. The average over all SNR levels is denoted AVG.

SNR (dB)	Noisy	LogFFT	NAT-LogFFT	FFT-MASK	NAT-FFT-MASK	NRM	NAT-NRM
20	3.150	3.257	3.240	3.380	3.303	3.502	3.449
15	2.843	2.973	2.952	3.098	3.017	3.227	3.165
10	2.539	2.679	2.653	2.805	2.728	2.936	2.867
5	2.260	2.384	2.353	2.502	2.432	2.628	2.558
0	1.754	2.103	2.068	2.196	2.137	2.317	2.248
-5	1.459	1.828	1.798	1.905	1.858	2.017	1.945
AVG.	2.426	2.537	2.511	2.648	2.579	2.771	2.705

The PESQ results for the static NAT systems in unseen noise are presented in Table 8.4. This time, a small, but insignificant, decrease in the PESQ scores at all SNR levels is obtained by the use of static NAT in all models. On average, the LogFFT scores had the least decline, while the FFT-MASK and NRM scores had a greater, and about equal, decline.

The STOI results for the static NAT systems in unseen noise are presented in Table 8.5. There is a similar small, but insignificant, decrease in the the STOI scores at all SNR levels for all the models. On average, NRM had the least decline, while the LogFFT and

Table 8.5: Average STOI scores for the different training targets and the static noise-aware training (NAT) models in unseen noise conditions. The average over all SNR levels is denoted AVG.

SNR (dB)	Noisy	LogFFT	NAT-LogFFT	FFT-MASK	NAT-FFT-MASK	NRM	NAT-NRM
20	0.958	0.963	0.961	0.964	0.962	0.970	0.970
15	0.921	0.932	0.929	0.936	0.933	0.947	0.946
10	0.866	0.886	0.882	0.893	0.887	0.911	0.907
5	0.790	0.823	0.816	0.833	0.824	0.858	0.852
0	0.698	0.745	0.734	0.755	0.744	0.787	0.780
-5	0.597	0.655	0.641	0.664	0.651	0.700	0.692
AVG.	0.805	0.834	0.827	0.841	0.834	0.862	0.858

FFT-MASK models had a greater, and about equal, decline.

The results in Tables 8.2 - 8.5 show that no tangible benefit is obtained by using static NAT with the noise prediction models. While there is a small increase in the objective metric scores in seen noise, this benefit is offset by the similar decrease in the objective metric scores in unseen noise. In addition, the change in scores is too small to be perceptually significant. As such, the changes do not justify the additional complexity that would be brought on by using static NAT.

8.4.2 Dynamic NAT Evaluation

In order to facilitate the discussion of the performance of the different systems, we introduce the following nomenclature: the dynamic NAT systems that use the NRM as a noise estimate, i.e., the Raw NRM Output columns in Tables 8.6 - 8.9, will be referred to with the prefix “raw”, whereas those that use an estimate of the added noise magnitude spectrum, i.e., the NRM-based Magnitude Spectrum columns in Tables 8.6 - 8.9, will be discussed using the prefix “MS (Magnitude Spectrum)”.

The PESQ results for the dynamic NAT systems in seen noise are presented in Table 8.6. The scores of the NRM-FMSK model are uniformly lower than those of the baseline

FFT-MASK (FMSK) model at all SNR levels. The difference is slight, with an average difference of 0.05 over all SNR levels. In contrast, the the PESQ scores of MS-FMSK are slightly better than the baseline FMSK model with an average difference of 0.08 in the PESQ scores. The two dynamic NAT NRM models, on the other hand, are virtually identical to the baseline NRM model. The average PESQ score of the raw-NRM system is identical to that of the baseline system, NRM, and the average score of the MS-NRM system is lower by 0.01.

Table 8.6: Average PESQ scores for the different training targets and the dynamic noise-aware training (NAT) models in seen noise conditions. FFT-MASK is denoted as FMSK, and the average over all SNR levels is denoted AVG.

SNR (dB)	Noisy	FMSK	NRM	Raw NRM Output				NRM-based Magnitude Spectrum			
				DNAT-FMSK	Oracle FMSK	DNAT-NRM	Oracle NRM	DNAT-FMSK	Oracle FMSK	DNAT-NRM	Oracle NRM
20	2.994	3.755	3.818	3.705	3.972	3.818	4.099	3.792	4.069	3.815	4.061
15	2.668	3.549	3.650	3.486	3.770	3.651	3.975	3.612	3.921	3.644	3.917
10	2.347	3.295	3.446	3.230	3.512	3.447	3.830	3.379	3.730	3.433	3.738
5	2.043	3.007	3.201	2.948	3.217	3.202	3.667	3.100	3.470	3.182	3.515
0	1.766	2.690	2.912	2.644	2.910	2.911	3.477	2.788	3.163	2.896	3.248
-5	1.459	2.342	2.581	2.309	2.592	2.580	3.247	2.440	2.844	2.577	2.948
AVG.	2.213	3.106	3.268	3.054	3.329	3.268	3.716	3.185	3.533	3.258	3.571

The STOI results for the dynamic NAT systems in seen noise are presented in Table 8.7. The scores of the raw-FMSK model are slightly better than those of the baseline model at the higher SNR levels of 10dB and above, but are slightly lower at SNR levels below 10dB. The average difference of 0.003 is not significant. The scores of the MS-FMSK system, on the other hand are slightly better than those of the baseline NRM model at all SNR levels, however, the average difference of 0.003 is also small and not significant.

The PESQ results for the dynamic NAT systems in unseen noise are presented in Table 8.8. In contrast to the performance of the systems in seen noise evaluations, there are more significant differences in the performance of the baseline and dynamic NAT systems in unseen noise. The scores of the raw-FMSK model are slightly higher than those of the baseline FMSK model at all SNR levels and by 0.07 on average. The scores of the MS-

Table 8.7: Average STOI scores for the different training targets and the dynamic noise-aware training (NAT) models in seen noise conditions. FFT-MASK is denoted as FMSK, and the average over all SNR levels is denoted AVG.

SNR (dB)	Noisy	FMSK	NRM	Raw NRM Output				NRM-based Magnitude Spectrum			
				DNAT-FMSK	Oracle FMSK	DNAT-NRM	Oracle NRM	DNAT-FMSK	Oracle FMSK	DNAT-NRM	Oracle NRM
20	0.958	0.975	0.977	0.976	0.986	0.977	0.987	0.974	0.988	0.969	0.985
15	0.921	0.959	0.963	0.960	0.976	0.963	0.979	0.960	0.980	0.955	0.976
10	0.866	0.935	0.942	0.936	0.959	0.942	0.966	0.938	0.966	0.935	0.961
5	0.790	0.901	0.913	0.900	0.933	0.912	0.948	0.907	0.945	0.906	0.941
0	0.698	0.855	0.871	0.848	0.894	0.871	0.924	0.861	0.915	0.867	0.914
-5	0.597	0.791	0.812	0.779	0.841	0.811	0.894	0.794	0.869	0.811	0.877
AVG.	0.805	0.903	0.913	0.900	0.932	0.913	0.950	0.906	0.944	0.907	0.942

FMSK model are also higher than than those of the baseline FMSK model. The difference is more significant: there is a difference of at least 0.1 in the scores at each SNR level, and a difference of about 0.12 on average.

The performance of the raw-NRM model, on the other hand, is virtually identical to that of the baseline NRM model with an average PESQ difference of 0.004, however, the performance of the MS-NRM model is significantly better than that of the baseline NRM model. The average PESQ difference is about 0.09, and the difference in PESQ scores increases at lower SNR levels, which is desirable.

Table 8.8: Average PESQ scores for the different training targets and the dynamic noise-aware training (NAT) models in unseen noise conditions. FFT-MASK is denoted as FMSK, and the average over all SNR levels is denoted AVG.

SNR (dB)	Noisy	FMSK	NRM	Raw NRM Output				NRM-based Magnitude Spectrum			
				DNAT-FMSK	Oracle FMSK	DNAT-NRM	Oracle NRM	DNAT-FMSK	Oracle FMSK	DNAT-NRM	Oracle NRM
20	3.150	3.380	3.502	3.442	3.941	3.503	4.120	3.487	4.100	3.565	4.062
15	2.843	3.098	3.227	3.163	3.723	3.229	4.008	3.212	3.966	3.309	3.909
10	2.539	2.805	2.936	2.876	3.448	2.938	3.874	2.923	3.782	3.026	3.713
5	2.260	2.502	2.628	2.574	3.145	2.632	3.713	2.620	3.535	2.723	3.467
0	1.754	2.196	2.317	2.271	2.837	2.321	3.516	2.314	3.241	2.414	3.185
-5	1.459	1.905	2.017	1.985	2.526	2.023	3.278	2.015	2.923	2.113	2.889
AVG.	2.426	2.648	2.771	2.719	3.270	2.775	3.751	2.762	3.591	2.858	3.538

The STOI results for the dynamic NAT systems in unseen noise are presented in Table 8.9. The performance of the baseline FMSK and NRM models is better than that of their corresponding dynamic NAT models. The scores of the baseline FMSK model are higher than those of the raw-FMSK model at all SNR levels and by about 0.05 on average. Similarly, the scores of the baseline FMSK model are higher than those of the MS-FMSK model by about 0.04 on average, and at all SNR levels. The baseline NRM model also outperforms both the raw-NRM and MS-NRM models at all SNR levels, and the difference in scores averaged over all SNR levels is about 0.05 in both cases.

Table 8.9: Average STOI scores for the different training targets and the dynamic noise-aware training (NAT) models in unseen noise conditions. FFT-MASK is denoted as FMSK, and the average over all SNR levels is denoted AVG.

SNR (dB)	Noisy	FMSK	NRM	Raw NRM Output				NRM-based Magnitude Spectrum			
				DNAT-FMSK	Oracle FMSK	DNAT-NRM	Oracle NRM	DNAT-FMSK	Oracle FMSK	DNAT-NRM	Oracle NRM
20	0.958	0.975	0.977	0.967	0.985	0.970	0.987	0.967	0.987	0.964	0.984
15	0.921	0.959	0.963	0.943	0.972	0.947	0.978	0.944	0.978	0.944	0.974
10	0.866	0.935	0.942	0.904	0.951	0.911	0.964	0.908	0.964	0.911	0.958
5	0.790	0.901	0.913	0.849	0.919	0.858	0.946	0.854	0.942	0.862	0.935
0	0.698	0.855	0.871	0.776	0.876	0.787	0.922	0.783	0.910	0.795	0.904
-5	0.597	0.791	0.812	0.687	0.822	0.701	0.893	0.696	0.865	0.713	0.865
AVG.	0.805	0.903	0.913	0.854	0.921	0.862	0.948	0.859	0.941	0.865	0.937

Tables 8.6 - 8.9 also show the performance of oracle models. The objective metric scores of the oracle models shows how the given model would have performed if the added noise signal spectrum was known. As such, these scores establish the upper limits on the performance of each of the trained system models. An examination of the oracle FMSK models in Table 8.6 shows that the performance of oracle MS-FMSK is better than that of oracle raw-FMSK, a performance difference that shows up in the actual or real models. Similarly, oracle MS-FMSK outperforms oracle raw-FMSK in Table 8.7, and the real MS-FMSK model was better than the baseline model while the real raw-FMSK was worse than the baseline system. Similar results can be observed in Tables 8.8 and 8.9 suggesting that the NRM is the better choice for implementing dynamic NAT when the FFT-MASK is the

training target.

A comparison of the NRM oracle systems in Table 8.6 is also interesting. Although oracle raw-NRM outperforms oracle MS-NRM, there is little difference in the real systems. In particular, oracle raw-NRM can be seen to be increasingly better than oracle MS-NRM at the lower SNR levels. A similar observation can be made from the performance in unseen noise, Table 8.8. This discrepancy might be related to the fact that the added noise representation used to augment the features in the raw-NRM systems is equivalent to the training target, the NRM (7.2). It also shows, however, that log magnitude spectral (MS) features of the added noise are a fairly robust choice for dynamic NAT when the NRM is the training target. The performance of the oracle NRM models in Tables 8.8 and 8.9 corroborate this claim. It can be seen the MS-NRM performs better that raw-NRM on both the PESQ and STOI metrics even though the reverse is the case with the oracle models.

Overall, the choice between the baseline and dynamic NAT models is more nuanced. One factor that should affect the choice of a model is the environment in which the model is to be deployed. If there is some knowledge of the type of interference to be encountered and the model can be trained with those noise types, the baseline models are a suitable choice. If there is no knowledge of the interference, on the other hand, the dynamic NAT models should be considered for their superior performance in unseen noise conditions. Another factor that could affect the choice of a model is the amount of latency than can be tolerated. The dynamic NAT models would have higher latency as the noisy input features have to be processed by two neural networks, whereas, the noisy input features are only processed by a single neural network in the baseline models.

8.5 Conclusion

In this chapter, we investigated noise-aware training approaches for noise prediction DNN models. The driving force behind the use of noise-aware training was the need to further improve the performance of the models in unseen noise and in low SNR environments.

Static and dynamic NAT approaches were implemented and their performance was evaluated in seen and unseen noise conditions. The static NAT models under-performed their corresponding baseline models and proved to be unsuitable. The performance of the dynamic NAT models, on the hand, was about identical to that of the baseline models in seen noise conditions, but better than that of the baseline in enhancing speech quality in unseen noise conditions. Augmenting the input features with the noise ratio mask features was the better choice for FFT-MASK training targets, while using log magnitude spectral features was the better choice for the NRM target.

CHAPTER 9
A MASK-BASED POST PROCESSING APPROACH FOR IMPROVING THE QUALITY AND INTELLIGIBILITY OF DEEP NEURAL NETWORK ENHANCED SPEECH

This chapter presents a method for the post-processing of deep neural network (DNN) enhanced speech. The work presented in this chapter has been published in the *Proceedings of the 2017 International Conference On Machine Learning And Applications* [122].

9.1 Introduction

In the previous chapters, we studied different techniques to enhance speech with neural networks. In this chapter, we shift focus to another task, namely, the post-processing of the enhanced speech. While there has been an intense focus on neural network-based speech enhancement and several works have been published, there has been far less interest on the development and use of post processing techniques with neural networks. One reason for this development might be that such techniques are deemed unnecessary with data-driven models where no assumptions are made about the relationships between the speech and noise signals. Post processing techniques were commonly used with MMSE algorithms restore distortions introduced by overly-aggressive noise suppression rules. Techniques like harmonic generation [124, 125] and codebook-based processing [126] were shown to restore over-attenuated portions of the speech spectrum and improve the quality of Wiener filtered speech.

Recently, the DNN training procedure, which typically includes mean and variance normalization, was shown to result in an over-smoothed spectrum, and global variance equalization was proposed as a method to improve the quality of the enhanced speech [127]. Similarly, a mask-based post-processing method was shown to improve both the

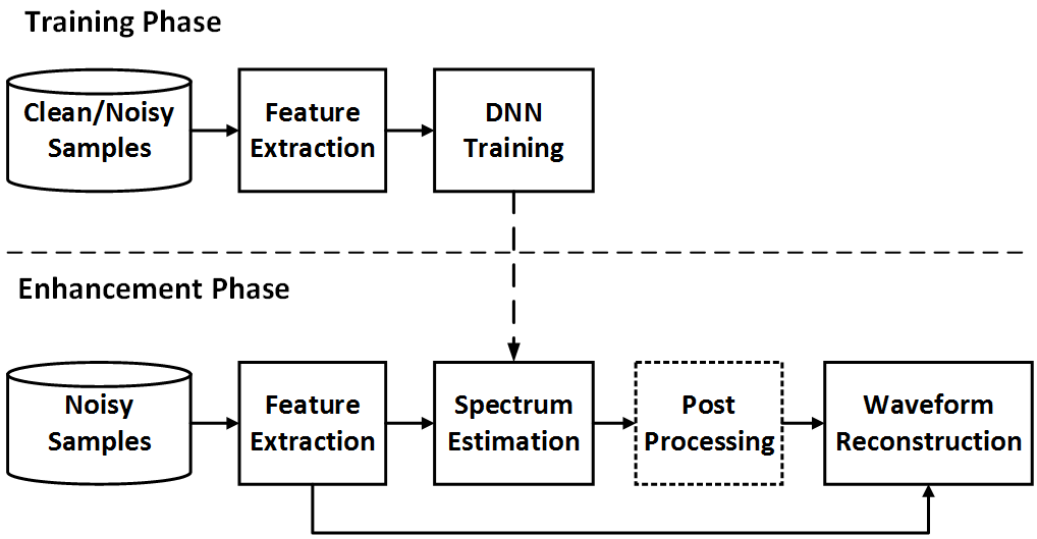


Figure 9.1: A block diagram of the baseline and proposed systems.

quality and intelligibility of DNN enhanced speech [128]. These works show that the use of post processing techniques with DNNs can be beneficial.

In this chapter, we present a post-processing method for DNN enhanced speech. The technique is a mask-based approach in which a time-frequency (T-F) weighting computed from the output of a well-trained neural network that predicts noise-free speech spectra from noisy input spectra is used to suppress T-F units that are dominated by noise sources [129]. The method is also simple, and does not require an expansion of the input or feature vectors, or further training, as it uses only the spectral estimates of the DNN. A series of experiments is presented to compare the performance of the proposed method with a baseline system under matched and mismatched noise conditions, and statistical analyses of the results are presented. The rest of the chapter is organized as follows: the baseline and proposed systems are described in Section 9.2, experiments are described in Section 9.3, results are presented in Section 9.4, discussions follow in Section 9.5, and conclusions are presented in Section 9.6.

9.2 System Overview

A block diagram of the baseline neural network system enhancement system along with the proposed modification is shown in Figure 9.1. Both systems use an identical training phase, but differ in the implementation of the enhancement phase. In the training phase, noisy and noise-free input-output log-spectra pairs are extracted from the framed training samples and used in training the neural network. The noisy log-spectra are fed into the neural network and the network learns a non-linear map between the noisy input and the noise-free output log-spectra.

The network is trained using the back-propagation algorithm to minimize a mean-square error criterion. Network parameters are updated using mini-batch stochastic gradient descent with momentum. The error criterion is

$$E = \frac{1}{N} \sum_{i=1}^{N} ||\hat{\mathbf{S}}_i(\mathbf{y}_i, \mathbf{\Theta}) - \mathbf{S}_i||^2 \tag{9.1}$$

where $\mathbf{y}_i$ is the input to the network, $\mathbf{\Theta} = \{\mathbf{W}, \mathbf{b}\}$, represents the weights and biases in the network, $\hat{\mathbf{S}}_i(\mathbf{y}_i, \mathbf{\Theta})$ is the output of the network, $\mathbf{S}_i$, the noise-free spectrum, is the desired target, and N is the mini-batch size.

In the enhancement phase of the baseline system, features extracted from noisy speech frames are fed into the trained network, and the network generates its best estimate of the corresponding noise-free spectra. The noise-free spectral estimates are combined with the noisy phase and the enhanced speech is synthesized using the overlap-add method [100]. In the proposed system, the network computes the noise-free spectral estimate in the same manner as in the baseline system. The noise-free spectral estimate is then used to compute a T-F mask. The mask is computed as

$$H(t, \omega) = \min \left\{ \left(\frac{\hat{S}^2(t, \omega)}{X^2(t, \omega)} \right)^{\frac{1}{2}}, 1 \right\}, \tag{9.2}$$

where $\hat{S}^2(t,\omega)$ represents the estimated signal power spectrum, and $X^2(t,\omega)$ represents the noisy signal power spectrum. The new post-processed noise-free spectral estimates are then obtained as

$$\hat{S}_{pp}(t,\omega) = H(t,\omega)X(t,\omega) \tag{9.3}$$

where $X(t,\omega)$ is the noisy speech complex spectrum. The overlap-add method is then used to reconstruct the enhanced speech waveform.

The T-F mask is computed by normalizing the estimated signal power by the noisy signal power and enforcing an upper bound. The mask thus represents a probability or confidence that a bin contains speech. The gain-limiting is important in reducing distortions caused by estimation errors and would be further discussed in Section 9.5.

9.3 Experiments

Table 9.1: Description of noise types used in testing.

Noise	Description	Noise	Description
n1	Crowd	n6	Water
n2	Machine	n7	Wind
n3	Alarm/Siren	n8	Bell
n4	Traffic/Car	n9	Cough
n5	Animal	n10	Clap

All experiments were performed using recorded sentences from the IEEE Corpus [98] included with the NOIZEUS database [8]. The corpus is comprised of 72 lists, each of which contains 10 sentences. Our noise samples came from a database of 100 non-speech sounds [99]. Both the noise free speech and noise recordings were resampled to 8kHz. The training dataset was comprised of sentences from lists 1 - 40, while testing was done with the 50 sentences from lists 68 - 72. Noisy utterances were created by adding 30 types of noise to each of the sentences at six SNR levels ranging from 20dB to -5dB in 5dB steps. This produced a training set of about 50 hours in length

Log power spectral features were extracted from the noisy and noise-free speech signals The speech signals were divided into 32ms frames. Fourier analysis was performed using a Hamming window. Input features were normalized to have zero mean and unit variance, while training targets were not normalized. To allow the networks to take advantage of temporal information, each input vector included adjacent time frames. The input feature vector was further expanded by supplying information about the noise present in each utterance. This method, which has proved beneficial in both speech enhancement and recognition, has been termed noise-aware training [6, 19]. Consequently, each input vector was constructed as

$$\mathbf{y}_i = [\mathbf{x}_{i-l}, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_{i+l}, \hat{\mathbf{n}}_i] \tag{9.4}$$

The noise estimate, $\hat{\mathbf{n}}_i$, was fixed for each utterance and obtained by averaging over the first few frames of noisy log-spectra.

$$\hat{\mathbf{n}}_i = \hat{\mathbf{n}} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{x}_k \tag{9.5}$$

Input frames with values of $l \in \{0, 5\}$, which respectively represent no temporal context and five context frames, for an input length of eleven frames not including the noise estimate, were utilized in training and evaluation, and the noise estimates were obtained from the first five frames of each noisy utterance.

Three network topologies were trained and evaluated. The first, NN_1, was a single hidden layer network with 6000 hidden nodes. The second and third, NN_2 and NN_3, were deep networks with two and three hidden layers respectively, each containing 2000 hidden nodes. These are respectively networks with increasing regression capability [19]. The hidden layers of all the networks used the rectified linear unit (ReLU) activation function [107, 108], and the output layers were linear. Weights and biases of all the layers were initialized following the method of He *et al*. [109], and the networks were trained using gradient descent with momentum. The initial learning rate was set to 0.001 for the first

10 epochs, then decreased by 10% every subsequent 10 epochs, and the the momentum coefficient was set for 0.9. A mini-batch size of 128 samples was used. The single hidden-layer network was trained for 30 epochs, while the deep networks were trained for 50 epochs. All networks were implemented and trained using the TensorFlow library [110].

Testing was done using both matched and mismatched noise types. Ten noise types were used in each of the testing scenarios. In matched noise tests, each of the noise types used during the enhancement or evaluation phase was one of the noise types used during the training phase. Conversely, in mismatched noise testing, each of the noise types used during the evaluation phase had not been previously seen by the neural network. A description of the noise types, which include several non-stationary noises, is listed in Table 9.1.

Speech quality and intelligibility were objectively evaluated using the perceptual evaluation of speech quality (PESQ) [101] and short-time objective intelligibility (STOI) [118] metrics respectively. PESQ scores range from -0.5 to 4.5 while STOI scores range from 0 to 1. These measures have been shown to have high correlation with subjective listening tests [119, 120] .

Statistical analyses of the results were carried out in order to determine if there were statistically significant differences between the objective metric scores of the baseline and proposed systems. This analysis included the one-way Analysis of Variance (ANOVA) and Tukey's HSD (honest significant difference) tests [130, 131]. The one-way ANOVA test was used to determine whether any of the mean metrics scores were significantly different, and the Tukey HSD test was used as a post-hoc test to determine which of particular means were significantly different. The Tukey HSD test is suitable here as it does not lose significance when multiple comparisons are performed [130].

9.4 Results

9.4.1 Evaluation in Matched Noise

The PESQ results for the different models are presented in Table 9.2. The use of the T-F mask for post processing mask clearly improves the performance of each of the baseline DNN models. The shallow network, NN_1, has the largest performance boost, while NN_2 and NN_3 have smaller gains in performance. In addition, there is greater improvement at higher SNR values than at lower SNR values. There is, however, improvement at all SNR values.

The STOI results presented in Table 9.3 also show improvement of the post-processed models over the baseline models. At the highest SNR level of 20dB, the baseline models fail to improve the intelligibility of the input signal even though there are improvements in speech quality. The use of T-F mask for post processing, however, increases both the quality and intelligibility. As observed with the PESQ scores, there are greater performance improvements at the higher SNR values, however, the reduction in performance with SNR is not as pronounced as with the PESQ scores. Overall, PP_2 provides the best performance.

Table 9.2: Average PESQ scores for the baseline and proposed systems in matched noise. NN_x represent a network with x hidden layers and PP_x is the corresponding version of the proposed system. The average over all SNR levels is denoted AVG.

SNR (dB)	Noisy	NN_1	PP_1	NN_2	PP_2	NN_3	PP_3
20	3.03	3.53	3.74	3.72	3.84	3.70	3.79
15	2.70	3.44	3.60	3.61	3.70	3.58	3.65
10	2.38	3.32	3.43	3.47	3.53	3.44	3.50
5	2.07	3.15	3.23	3.30	3.34	3.29	3.32
0	1.79	2.94	3.00	3.09	3.12	3.10	3.12
-5	1.50	2.67	2.72	2.84	2.85	2.87	2.88
AVG	2.25	3.18	3.29	3.34	3.40	3.33	3.38

Table 9.3: Average STOI scores for the baseline and proposed systems in matched noise.

SNR (dB)	Noisy	NN_1	PP_1	NN_2	PP_2	NN_3	PP_3
20	0.96	0.92	0.96	0.96	0.98	0.96	0.97
15	0.93	0.91	0.95	0.95	0.97	0.95	0.96
10	0.87	0.90	0.94	0.94	0.95	0.94	0.95
5	0.80	0.88	0.91	0.92	0.93	0.92	0.93
0	0.71	0.85	0.88	0.89	0.90	0.89	0.90
-5	0.61	0.80	0.83	0.85	0.86	0.86	0.86
AVG	0.81	0.88	0.91	0.92	0.93	0.92	0.93

9.4.2 Evaluation in Mismatched Noise

Table 9.4: Average PESQ scores for the baseline and proposed systems in mismatched noise.

SNR (dB)	Noisy	NN_1	PP_1	NN_2	PP_2	NN_3	PP_3
20	3.18	3.35	3.52	3.46	3.53	3.48	3.54
15	2.87	3.18	3.28	3.25	3.29	3.26	3.30
10	2.57	2.96	3.02	3.00	3.02	3.02	3.04
5	2.29	2.70	2.73	2.73	2.74	2.76	2.77
0	2.04	2.41	2.43	2.45	2.45	2.48	2.49
-5	1.78	2.11	2.12	2.15	2.15	2.20	2.20
AVG	2.45	2.79	2.85	2.84	2.87	2.87	2.89

The PESQ results for the different models in mismatched noise are presented in Table 9.4. The results follow similar trends to the results in matched noise shown in Table 9.2. One noticeable difference, however, is that at mid and high input SNR, each of the proposed, i.e. PP models is equivalent or better in performance than the corresponding deeper baseline model. For example, PP_1 is better than NN_2, and PP_2 is better than NN_3.

The STOI results for the different models in mismatched noise are presented in Table 9.5. It can be observed that the PP models are always outperform their corresponding

Table 9.5: Average STOI scores for the baseline and proposed systems in mismatched noise.

SNR (dB)	Noisy	NN_1	PP_1	NN_2	PP_2	NN_3	PP_3
20	0.96	0.90	0.96	0.95	0.97	0.95	0.97
15	0.92	0.89	0.94	0.94	0.95	0.94	0.95
10	0.88	0.86	0.91	0.91	0.92	0.91	0.92
5	0.81	0.82	0.86	0.86	0.88	0.87	0.88
0	0.74	0.76	0.79	0.80	0.81	0.81	0.82
-5	0.65	0.68	0.71	0.72	0.73	0.73	0.73
AVG	0.83	0.82	0.86	0.86	0.88	0.87	0.88

baseline models. It can also be observed that while even the deep baseline models slightly impair the intelligibility of the noisy signal that the highest SNR level, the deep PP models improve the intelligibility at all SNR levels. In addition, PP_1 performs as well or better than NN_2 at mid and high SNR, and PP_2 outperforms NN_3 at all input SNR levels. Hence, we can say that the use of post processing mask produces a "gain" of one hidden layer. Overall, PP_3 performs the best in mismatched noise.

9.4.3 Statistical Comparison of Models

We obtained p-values of 7.62×10^{-134} and 0 when the one-way ANOVA procedure was carried out on the matched noise PESQ and STOI results respectively. Similarly, p-values of 1.422×10^{-10} and 1.59×10^{-132} were obtained when the one-way ANOVA procedure was carried out on the mismatched noise PESQ and STOI results respectively. The small p-values indicates that we can reject the null hypothesis that all the models have the same efficacy in enhancing speech. A subset of the matched noise model comparisons using the Tukey HSD test is shown in Table 9.6. The table shows the models being compared in each row, the difference of means, $\Delta\mu = \mu_{\mathrm{NN}_i} - \mu_{\mathrm{PP}_i}$, and the p-value of the test statistic. From the small p-values, it can be inferred that the difference between the models is statistically significant [130]. Consequently, it can be concluded that using the post-processing

method improves both the quality and intelligibility of enhanced speech in matched noise conditions.

Table 9.6: Statistical comparison of the objective metric scores for the baseline and proposed models in matched noise.

Models		PESQ		STOI	
		$\Delta\mu$	p-Value	$\Delta\mu$	p-Value
NN_1	PP_1	-0.109	2.07×10^{-8}	-0.036	2.07×10^{-8}
NN_2	PP_2	-0.056	2.07×10^{-8}	-0.015	2.07×10^{-8}
NN_3	PP_3	-0.047	3.32×10^{-5}	-0.013	2.07×10^{-8}

Table 9.7: Statistical comparison of the objective metric scores for the baseline and proposed models in mismatched noise.

Models		PESQ		STOI	
		$\Delta\mu$	p-Value	$\Delta\mu$	p-Value
NN_1	PP_1	-0.065	2.63×10^{-4}	-0.042	2.07×10^{-8}
NN_2	PP_2	-0.025	0.560	-0.014	5.32×10^{-8}
NN_3	PP_3	-0.020	0.795	-0.011	4.02×10^{-4}

A subset of the mismatched noise model comparisons is shown in Table 9.7. The p-values of the PESQ results show that the difference between models NN_1 and PP_1 is statistically significant, but the differences between NN_2 and PP_2 and NN_3 and PP_3 are not. The small p-values of the STOI results, on the other hand, show that the difference between the models is statistically significant. Consequently, it can be concluded that using the post-processing method always improves the intelligibility of enhanced speech in mismatched noise conditions.

9.5 Discussion

To investigate the impact of post-processing, we examine the level of distortion in the magnitude spectrum of the enhanced speech. Two types of distortion, attenuation and amplification, are of concern. Attenuation distortions resulting in a loss of speech spectral

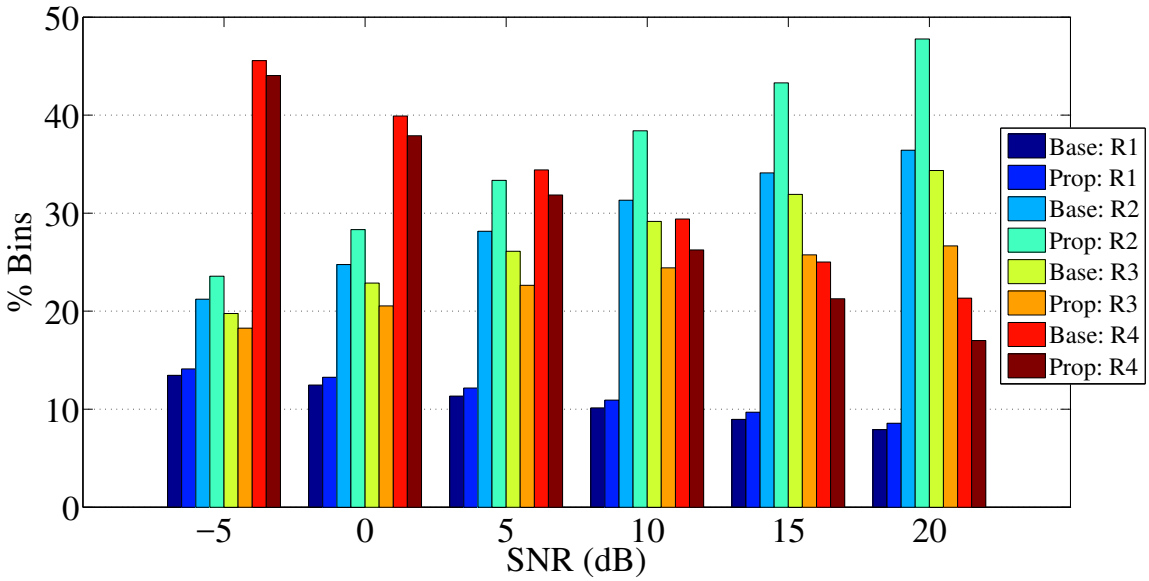


Figure 9.2: Magnitude spectrum bin distribution for the baseline and proposed systems.

components, and amplification distortions resulting in undesirable audible artifacts would both adversely affect speech quality. In addition, severe amplification distortions could adversely affect intelligibility as these were shown to be related to the signal-to-residual spectrum ratio which is highly correlated with speech intelligibility [10]. We examine four regions of distortion:

Region 1 (R1) In this region, $\hat{S}(k) < 0.5S(k)$, representing severe attenuation of more than 6dB SNR i.e. $\text{SNR}_{ENH}(k) < \text{SNR}(k) - 6.02\text{dB}$, where $\text{SNR}_{ENH}(k)$ and $\text{SNR}(k)$ represent the enhanced and true signal SNR respectively.

Region 2 (R2) In this region, $0.5S(k) \leq \hat{S}(k) < S(k)$, representing milder attenuation.

Region 3 (R3) In this region, $S(k) \leq \hat{S}(k) < 2S(k)$, representing milder amplification.

Region 4 (R4) In this region, $\hat{S}(k) > 2S(k)$, representing severe amplification distortion of more than 6dB SNR.

The percentage distribution of spectral bins in these four regions for enhanced speech obtained with both the baseline and proposed systems, NN_3 and PP_3, in mismatched noise

is shown in Figure 9.2. The displayed results were obtained using all the bins in the entire test sample with non-zero magnitude.

The baseline system is characterized with increased Region 4 distortion at lower SNR values. The mask-based post processing tends to reduce the Region 3 and Region 4 distortion while increasing the Region 1 and Region 2 distortion. The increase in Region 1 distortion is slight, occurring in only about 0.7% of the bins on average. In addition, there is a greater reduction in Region 4 distortions at higher SNR values. This suggests that the post processing reduces the presence of spurious spectral peaks or artifacts at the cost of a tendency to slightly over-attenuate parts of the speech spectrum. However, this trade-off is known to improve intelligibility [10] and can be inferred to improve perceived speech quality.

9.6 Conclusion

In this chapter, we proposed a post processing approach for DNN-based speech enhancement systems. The method is simple, does not require additional training or expansion of the feature or target vectors, and can be viewed as a mask-based approach in which a noisy speech signal is processed by a time-frequency (T-F) weighting derived from the noise-free spectral estimate of a DNN. Objective test results from experimental evaluations in matched and mismatched noise conditions showed that the proposed approach always resulted in improved speech quality and intelligibility, and it always outperformed a corresponding baseline system without post processing. Further analysis of the objective test scores showed that the speech quality and intelligibility improvements in matched noise, and the intelligibility improvements in mismatched noise were statistically significant.

A comparison of enhanced speech samples from the baseline systems and the proposed systems showed that post processing reduced the incidence of severe amplification distortion but also slightly increased the incidence of over-attenuation. As such, a trade-off between amplification and attenuation distortions to be directly connected to the observed

speech quality and intelligibility improvements.

CHAPTER 10
CONCLUSION

This concluding chapter summarizes the main findings and contributions of this dissertation. First, the contributions are enumerated and discussed, and then suggestions for further work are given.

10.1 Contributions

In this dissertation, we studied certain aspects of the development of an efficient neural network-based speech enhancement system. The main findings and contributions are as follows:

In Chapter 3, we developed a framework for speech enhancement using the extreme learning machine (ELM). We showed for a spectral mapping ELM that a min-max normalization scheme, in which input features are normalized to the range [-1,1], produced the lowest prediction error. We also showed that there was no benefit from regularizing the weights in the multi-output regression model used for ELM-based speech enhancement. We showed that the use of temporal context was beneficial; however, only a single context frame, i.e., a total input length of three frames, was optimal. We then showed that the use of larger networks and larger training datasets increased the quality of the enhanced speech, and the spectral mapping ELM surpassed the optimally-modified log spectral amplitude (OM-LSA) estimator with as little as 5 hours of training data in matched noise, and 10 hours of training data in mismatched or unseen noise.

In addition, we studied a T-F mask-based ELM framework and showed the observations from employing temporal context were not contradictory: in the ELM enhancement framework, the size of the network must be increased as the number of context frames is increased in order to take advantage of the increased contextual information. Finally, we

showed the T-F mask-based ELM framework was superior. Its performance was consistent over a wide range of training data sizes, and it outperformed the OM-LSA algorithm in matched and mismatched noise with just 1 hour of training data.

In Chapter 4, we developed two approaches to increase the prediction accuracy of the batch ELM in multivariate regression tasks. These algorithms, just like the ELM, do not require iterative tuning of the weights in the network, or repeated processing of the training dataset. We then showed that these algorithms were always more accurate that the original ELM algorithm in evaluations on a synthetic dataset and multiple real-world datasets.

Chapter 5 brought together and completed our study of the ELM for speech enhancement. One of the improved ELM algorithms, the canonical ELM, was used for speech enhancement, and its performance and that of the original ELM algorithm were compared using objective performance metrics. We then compared the performance of the canonical ELM to that of neural networks trained conventionally with the back propagation algorithm.

In Chapter 6, we developed noise a noise prediction and time-domain subtraction approach to speech enhancement. First, we developed an extension of the overlap-add procedure to allow for real-time synthesis of enhanced speech from the DNN's estimates of the noise spectral magnitude. We then developed a variation of the noise prediction architecture that estimated the magnitude spectrum of the noise-free speech using spectral subtraction. We compared the performance of the time-domain and spectral subtraction approaches and showed there was no perceptual difference between these two models. As such, no advantage was gained by reconstructing the noise instead of the enhanced speech with the noisy phase. We then showed the noise prediction system outperformed a conventional speech prediction system on speech quality metrics at high SNR levels in seen noise tests, but under-performed the same system at low SNR levels, and in unseen noise tests. In addition, we showed that the noise prediction systems exhibited strong performance on intelligibility metrics at high SNR levels in both seen and unseen noise tests.

In Chapter 7, we further developed the noise prediction and time-domain subtraction approach to speech enhancement by studying the performance of three different training targets. We introduced a new training target, the noise ratio mask (NRM), which is equivalent to square-root Wiener filter if the additive noise and speech are assumed to be uncorrelated, and the noise is considered the desired signal. We then showed that in the noise prediction architecture, the mask-based targets were superior to the spectral target. The NRM, in particular, was the best training target. It outperformed a benchmark speech prediction model, the noise-aware trained model, in improving speech quality and intelligibility in seen noise conditions, and in improving intelligibility in unseen noise, but was slightly worse in enhancing speech quality than the NAT model at lower SNR values.

We continued our development of the noise prediction architecture in Chapter 8 by examining the use of noise-aware training (NAT) strategies as a means of improving the robustness of the noise prediction networks. We showed that while the static NAT models were marginally better than their corresponding baseline models in seen noise types, they under-performed the same baseline models in unseen noise types and were, therefore, unsuitable. We then showed that a noise prediction network could be used to provide noise estimates in implementations of dynamic NAT strategies. In particular, we cascaded noise prediction networks and showed that the performance of the FFT-MASK model could be improved by augmenting its input features with the output of a network with an NRM target. We also showed the performance of the NRM model could be improved by augmenting its input features with magnitude spectral features derived from the output of a network with an NRM target.

In Chapter 9, we presented an analysis of a post processing approach for improving the quality and intelligibility of DNN enhanced speech. We had earlier applied this method in the development of the noise prediction architecture in Chapter 6 and Chapter 7 and now applied it to conventional speech prediction networks. We showed that the proposed approach always resulted in improved speech quality and intelligibility, and it always outperformed

a corresponding baseline system without post processing. Furthermore, the speech quality and intelligibility improvements in matched noise, and the intelligibility improvements in mismatched noise were statistically significant.

10.2 Suggestions for Future Work

In this dissertation, we studied the use of the extreme learning machine for the large scale task of speech enhancement and developed a a noise prediction and time-domain subtraction approach speech enhancement. Since both of these approaches are seminal, there are several extensions that naturally follow. We discuss some of these in the section.

- *Evaluation of other ELM networks*

  In this dissertation, we used the ELM algorithm to train single hidden layer networks; however, the results in Chapter 3 showed the benefits of larger networks, with more degrees of freedom, especially when using adjacent frames to provide temporal context. Deep networks, that are able to learn multiple layers of abstraction [33], have been shown to be more effective on several learning tasks including speech enhancement [19, 132]. An evaluation of deep ELM models such as the multilayer ELM [133], hierarchical ELM [75], and the deep representations learning ELM [134] should therefore be performed, as these models could prove to more effective than the single hidden layer model that was studied. The use of deep models could also allow the use of a smaller number of hidden layer nodes which would reduce training time. Another ELM model worth investigating is the fully complex ELM [135]. This ELM model could allow for the representation of both the magnitude and phase of the noisy and noise-free speech, and could, therefore, prove effective in speech enhancement.

- *Further development of a multivariate extreme learning machine*

  Two different approaches, the canonical ELM and the two-stage ELM, were shown

to improve the prediction accuracy of the ELM on multivariate datasets. The algorithmic approach of the canonical ELM could be combined with the data-driven approach of the two-stage ELM such that the two-stage ELM is trained using maximally correlated data. In addition, the use of non-random techniques to shape input weights [136], or the use of restricted Boltzmann machines to determine the input weights [137] should be investigated as a means to further improve the performance of the canonical ELM. Lastly, the two-stage ELM should be used for speech enhancement and its performance compared to that of the ELM and canonical ELM.

- *Further improvements of the noise prediction and time domain subtraction framework*

  The noise prediction architecture was newly introduced in this dissertation, therefore, there are several avenues for further investigations. First, the study of training targets in Chapter 7 demonstrated the impact of the choice of training target on system performance. Further investigation of training targets should thus be carried out. In particular, phase-aware training targets [138], which utilize objective functions that include both amplitude and phase error, and complex ratio masking targets, both of which attempt to enhance both the magnitude and phase of the noisy speech are some suitable targets for future exploration.

  Another area of investigation that could yield big dividends is the development of perceptual or psychoacoustic noise prediction neural network models. Such models would employ features that target the only audible noise and could, therefore, result in improved performance. As mentioned in Chapter 2, perceptual variants of classical speech enhancement algorithms like spectral subtraction and Wiener filtering yielded improvements over the original methods. In addition to the use of perceptual-based features, neural networks for speech enhancement can also be trained to maximize perceptual-based objective functions [139]. The use of perceptual-based features and/or objective criteria could help improve the low-SNR performance of the noise

prediction models.

Lastly, other network architectures including recurrent networks like the long-short term memory (LSTM) [140] and generative adversarial networks (GANs) [141] should be used in the noise prediction architecture. In addition, end-to-end systems, that work with the raw audio and, consequently, do not require a choice of suitable input features or training targets should be also be evaluated for the noise prediction framework. An example of an end-to-end system is the speech enhancement generative adversarial network (SEGAN) [142].

- *Further improvements through the use of signal processing techniques*

  Speech enhancement involves modification of spectral magnitudes by means of implied or direct multiplication (in the case of T-F masking) in the frequency domain. As is well known, multiplication in the frequency domain corresponds to convolution in the time domain, and certain conditions must be fulfilled in order to obtain linear and not circular convolution [143]. The linear filtering view of STFT processing has been largely ignored in machine learning approaches to speech enhancement, and the lack of attention to this detail could result in the presence of time-aliasing distortions and audible artifacts in the enhanced speech. The use of techniques designed to reduce, or entirely eliminate, distortions should, therefore, be investigated. Some of these approaches include "brick-wall" windowing, [144] and "artifact-free" convolution methods [145] such as Fast Fourier Transform (FFT) convolution by frequency extension and FFT convolution by frequency splitting.

Appendices

APPENDIX A

DESCRIPTION OF THE MULTIVARIATE DATASETS

This appendix provides a detailed description of the multivariate datasets utilized in Chapter 4. Further details on the construction of any of the datasets or of the input and target attributes can be obtained from the publications of the original authors.

- WQ

  The Water Quality dataset [146] is concerned with the prediction of physical and chemical parameters of river water quality from biological parameters. It is comprised of 14 input attributes which describe the density of biological taxa and 16 target attributes that describe the measured values of physical and chemical water quality parameters.

- ATP

  The Airline Ticket Price datasets [147] are concerned with the prediction of airline ticket prices. The data consists of daily price quotes from a travel search website for 7 different origin-destination pairs. The target variables are either the next day ticket price (ATP1D), or the minimum price observed over the subsequent 7 days (ATP7D), for 6 flight preferences. The input variables include the number of days between the observation and departure dates, day of the week, pricing information, number of quotes from the all the airlines, and the category of flight - non-stop, one-stop, and two-stop flights. The entire feature set consists of 411 input variables which includes a mixture of Boolean, categorical, and numerical variables. The entire details on the construction of the datasets can be found in [147].

- EDM

  The Electrical Discharge Machining dataset [148] has two discrete target variables

that model the actions of a human operator in an electrical machining process. The operator controls the gap and flow in the machining process, and the goal is to be able to minimize machining time by learning and automatically reproducing the actions of a skilled operator. The input attributes are continuous and consist of the mean values and deviations of observed quantities of machining parameters monitored by the operator during the machining process.

- ENB

  The Energy Building dataset [149] is concerned with the the prediction of the heating load and cooling load of residential building as a function of eight input variables, namely, the relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area, glazing area distribution.

- SCM

  The Supply Chain Management dataset [150] contains 16 regression targets, each of which corresponds to the next day mean price (SCM1D) or the mean price for 20 days in the future (SCM20D) for each product in a supply chain simulation. The input variables are the observed prices and time-delayed (1,2, 4, 8 days delay) of each of the products.

- SLUMP

  The Slump Concrete dataset [151] is concerned with the prediction of three properties of concrete, namely, the slump, flow, and compressive strength, as a function of the content per unit volume of seven ingredients: cement, fly ash, blast furnace slag, water, superplasticizer, coarse aggregate, and fine aggregate.

REFERENCES

[1] J. Benesty, S. Makino, and E. J. Chen, Eds., *Speech Enhancement*, ser. Signals and Communication Technology. Springer, 2005.

[2] I. Rodomagoulakis, P. Giannoulis, Z. I. Skordilis, P. Maragos, and G. Potamianos, "Experiments on far-field multichannel speech processing in smart homes," in *2013 18th International Conference on Digital Signal Processing (DSP)*, Jul. 2013, pp. 1–6.

[3] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent neural networks for noise reduction in robust ASR," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[4] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, Apr. 2014.

[5] X. Zhao, Y. Wang, and D. Wang, "Robust speaker identification in noisy and reverberant conditions," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 4, pp. 836–845, 2014.

[6] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, IEEE, 2013, pp. 7398–7402.

[7] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR," in *International Conference on Latent Variable Analysis and Signal Separation*, Springer, 2015, pp. 91–99.

[8] P. C. Loizou, *Speech Enhancement: Theory and Practice*. CRC press, 2013, ISBN: 1466504218.

[9] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[10] P. C. Loizou and G. Kim, "Reasons why current speech-enhancement algorithms do not improve speech intelligibility and suggested solutions," *IEEE transactions on audio, speech, and language processing*, vol. 19, no. 1, pp. 47–56, 2011.

[11] N. Li and P. C. Loizou, "Factors influencing intelligibility of ideal binary-masked speech: Implications for noise reduction," *The Journal of the Acoustical Society of America*, vol. 123, no. 3, pp. 1673–1682, 2008.

[12] M. Parchami, W. P. Zhu, B. Champagne, and E. Plourde, "Recent developments in speech enhancement in the short-time Fourier transform domain," *IEEE Circuits and Systems Magazine*, vol. 16, no. 3, pp. 45–77, 2016.

[13] M. Brandstein and D. Ward, Eds., *Microphone Arrays: Signal Processing Techniques and Applications*, ser. Digital Signal Processing. Springer, 2001.

[14] Business Wire, *Amazon makes the high-performance 7-mic voice processing technology from amazon echo available to third-party device makers*, Web Page, [Online; accessed 20-March-2018].

[15] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.

[16] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[17] ——, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 2, pp. 443–445, 1985.

[18] I. Cohen, "On speech enhancement under signal presence uncertainty," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 1, 2001, 661–664 vol.1, ISBN: 1520-6149.

[19] Y. Xu, J. Du, L. R. Dai, and C. H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

[20] L. Hong, J. Rosca, and R. Balan, "Independent component analysis based single channel speech enhancement," in *Signal Processing and Information Technology, 2003. ISSPIT 2003. Proceedings of the 3rd IEEE International Symposium on*, IEEE, 2003, pp. 522–525, ISBN: 0780382927.

[21] K. W. Wilson, B. Raj, and P. Smaragdis, "Regularized non-negative matrix factorization with temporal dependencies for speech denoising," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[22] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2140–2151, 2013.

[23] B. Xia and C. Bao, "Speech enhancement with weighted denoising auto-encoder," in *INTERSPEECH*, 2013, pp. 3444–3448.

[24] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *Interspeech*, 2013, pp. 436–440.

[25] Y. Zhao, D. Wang, I. Merks, and T. Zhang, "DNN-based enhancement of noisy and reverberant speech," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6525–6529.

[26] K. Osako, R. Singh, and B. Raj, "Complex recurrent neural networks for denoising speech signals," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on*, 2015, pp. 1–5.

[27] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *ISCAS*, 2010, pp. 253–256.

[28] B. B. Le Cun, J. S. Denker, D Henderson, R. E. Howard, W Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, Citeseer, 1990.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[30] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. N. Sainath, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[31] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks.," in *Aistats*, vol. 9, 2010, pp. 249–256.

[32] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural computation*, vol. 22, no. 12, pp. 3207–3220, 2010.

[33] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[34] W. F. Schmidt, M. A. Kraaijveld, and R. P. Duin, "Feedforward neural networks with random weights," in *Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*, IEEE, 1992, pp. 1–4, ISBN: 0818629150.

[35] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[36] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, no. 34, p. 13, 2001.

[37] B. Schrauwen, D. Verstraeten, and J. Van Campenhout, "An overview of reservoir computing: Theory, applications and implementations," in *Proceedings of the 15th European Symposium on Artificial Neural Networks. p. 471-482 2007*, 2007, pp. 471–482.

[38] T. Gao, J. Du, Y. Xu, C. Liu, L.-R. Dai, and C.-H. Lee, "Improving deep neural network based speech enhancement in low SNR environments," in *International Conference on Latent Variable Analysis and Signal Separation*, Springer, 2015, pp. 75–82.

[39] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'79.*, IEEE, vol. 4, 1979, pp. 208–211.

[40] P Lockwood, J Boudy, and M Blanchet, "Non-linear spectral subtraction (NSS) and hidden markov models for robust speech recognition in car noise environments," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, IEEE, vol. 1, 1992, pp. 265–268.

[41] S. Kamath and P. Loizou, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise.," in *ICASSP*, Citeseer, vol. 4, 2002, pp. 44 164–44 164.

[42] Y. Lu and P. C. Loizou, "A geometric approach to spectral subtraction," *Speech communication*, vol. 50, no. 6, pp. 453–466, 2008.

[43] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series*. MIT press Cambridge, 1949, vol. 2.

[44] J. Lim and A. Oppenheim, "All-pole modeling of degraded speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 3, pp. 197–210, 1978.

[45] O. Cappé, "Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 345–349, 1994.

[46] J. H. Hansen and M. A. Clements, "Constrained iterative speech enhancement with application to speech recognition," *IEEE Transactions on Signal Processing*, vol. 39, no. 4, pp. 795–805, 1991.

[47] T. Sreenivas and P. Kirnapure, "Codebook constrained Wiener filtering for speech enhancement," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 383–389, 1996.

[48] R. Martin, "Speech enhancement using MMSE short time spectral estimation with gamma distributed speech priors," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 1, IEEE, 2002, pp. I–253–I–256, ISBN: 0780374029.

[49] ——, "Speech enhancement based on minimum mean-square error estimation and supergaussian priors," *IEEE transactions on speech and audio processing*, vol. 13, no. 5, pp. 845–856, 2005.

[50] B. Chen and P. C. Loizou, "A Laplacian-based MMSE estimator for speech enhancement," *Speech communication*, vol. 49, no. 2, pp. 134–143, 2007.

[51] R. McAulay and M. Malpass, "Speech enhancement using a soft-decision noise suppression filter," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 2, pp. 137–145, 1980.

[52] D. Malah, R. V. Cox, and A. J. Accardi, "Tracking speech-presence uncertainty to improve speech enhancement in non-stationary noise environments," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 2, 1999, 789–792 vol.2, ISBN: 1520-6149.

[53] I. Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *IEEE Transactions on speech and audio processing*, vol. 11, no. 5, pp. 466–475, 2003.

[54] N. Virag, "Single channel speech enhancement based on masking properties of the human auditory system," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 2, pp. 126–137, 1999.

[55] B. C. Moore, *An Introduction to the Psychology of Hearing*. Brill, 2012.

[56] T. S. Gunawan, E. Ambikairajah, and J. Epps, "Perceptual speech enhancement exploiting temporal masking properties of human auditory system," *Speech communication*, vol. 52, no. 5, pp. 381–393, 2010.

[57] T. S. Gunawan and E. Ambikairajah, "On the use of simultaneous and temporal masking in noise suppression applications," in *Proc. of 11th Int. Conf. Speech Science and Technology*, 2006.

[58] D Tsoukalas, M Paraskevas, and J Mourjopoulos, "Speech enhancement using psychoacoustic criteria," in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, vol. 2, IEEE, 1993, pp. 359–362, ISBN: 0780374029.

[59] D. E. Tsoukalas, J. N. Mourjopoulos, and G. Kokkinakis, "Speech enhancement based on audible noise suppression," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 6, pp. 497–514, 1997.

[60] S. Gustafsson, P. Jax, and P. Vary, "A novel psychoacoustically motivated audio enhancement algorithm preserving background noise characteristics," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, IEEE, vol. 1, 1998, pp. 397–400.

[61] Y. Hu and P. C. Loizou, "A perceptually motivated approach for speech enhancement," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, pp. 457–465, 2003.

[62] I. Cohen, "On the decision-directed estimation approach of Ephraim and Malah," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, IEEE, vol. 1, 2004, pp. I–293.

[63] S. Tamura, "An analysis of a noise reduction neural network," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, 1989, 2001–2004 vol.3, ISBN: 1520-6149.

[64] S. Tamura and M. Nakamura, "Improvements to the noise reduction neural network," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, IEEE, 1990, pp. 825–828, ISBN: 1520-6149.

[65] F. Xie and D. Van Compernolle, "A family of MLP based nonlinear spectral estimators for noise reduction," in *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, vol. 2, IEEE, 1994, II/53–II/56 vol. 2, ISBN: 0780317750.

[66] H. B. D. Sorensen, "A cepstral noise reduction multi-layer neural network," in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, 1991, 933–936 vol.2, ISBN: 1520-6149.

[67] Y. Xu, J. Du, L. R. Dai, and C. H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2014.

[68] T Hastie, R Tibshirani, and J Friedman, *The elements of statistical learning: Data mining, inference, and prediction., 2nd edition*, 2009.

[69] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[70] S.-W. Fu, Y. Tsao, and X. Lu, "SNR-Aware convolutional neural network modeling for speech enhancement.," in *INTERSPEECH*, 2016, pp. 3768–3772.

[71] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[72] ——, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, IEEE, vol. 2, 2004, pp. 985–990.

[73] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.

[74] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 2, pp. 513–529, 2012.

[75] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 4, pp. 809–821, 2016.

[76] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural networks*, vol. 17, no. 6, pp. 1411–1423, 2006.

[77] B. Li, J. Wang, Y. Li, and Y. Song, "An improved on-line sequential learning algorithm for extreme learning machine," in *International Symposium on Neural Networks*, Springer, 2007, pp. 1087–1093.

[78] H. Borchani, G. Varando, C. Bielza, and P. Larraaga, "A survey on multioutput regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, 2015.

[79] L. Breiman and J. H. Friedman, "Predicting multivariate responses in multiple linear regression," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 59, no. 1, pp. 3–54, 1997.

[80] P. J. Brown and J. V. Zidek, "Adaptive multivariate ridge regression," *The Annals of Statistics*, pp. 64–74, 1980.

[81] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[82] A. J. Izenman, "Reduced-rank regression for the multivariate linear model," *Journal of multivariate analysis*, vol. 5, no. 2, pp. 248–264, 1975.

[83] A Van Der Merwe and J. Zidek, "Multivariate regression analysis and canonical variates," *Canadian Journal of Statistics*, vol. 8, no. 1, pp. 27–39, 1980.

[84] T. W. Anderson, *An introduction to multivariate statistical analysis*, 3rd, ser. Wiley series in probability and statistics. Hoboken, N.J.: Wiley-Interscience, 2003, xx, 721 p. ISBN: 0471360910.

[85] L. DAmbra and R. Lombardo, "Predicting multivariate responses in non-linear regression," *Bull. Intl Statistical Inst*, 1999.

[86] V. Vapnik, S. E. Golowich, A. Smola, *et al.*, "Support vector method for function approximation, regression estimation, and signal processing," *Advances in neural information processing systems*, pp. 281–287, 1997.

[87] V. N. Vapnik, *Statistical learning theory*, ser. Adaptive and learning systems for signal processing, communications, and control. New York: Wiley, 1998, xxiv, 736 p. ISBN: 0471030031 (acid-free paper).

[88] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, ACM, 1992, pp. 144–152.

[89] V. N. Vapnik, *The nature of statistical learning theory*. New York: Springer, 1995, xv, 188 p. ISBN: 0387945598 (New York alk. paper).

[90] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[91] F. Pérez-Cruz, G. Camps-Valls, E. Soria-Olivas, J. J. Pérez-Ruixo, A. R. Figueiras-Vidal, and A. Artés-Rodrguez, "Multi-dimensional function approximation and regression estimation," in *International Conference on Artificial Neural Networks*, Springer, 2002, pp. 757–762.

[92] M. Sánchez-Fernández, M. de-Prado-Cumplido, J. Arenas-Garca, and F. Pérez-Cruz, "SVM multiregression for nonlinear channel estimation in multiple-input multiple-output systems," *IEEE transactions on signal processing*, vol. 52, no. 8, pp. 2298–2307, 2004.

[93] D. Tuia, J. Verrelst, L. Alonso, F. Pérez-Cruz, and G. Camps-Valls, "Multioutput support vector regression for remote sensing biophysical parameter estimation," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 4, pp. 804–808, 2011.

[94] B. O. Odelowo and D. V. Anderson, "Speech enhancement using extreme learning machines," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, IEEE, Oct. 2017, pp. 200–204.

[95] ——, "A framework for speech enhancement using extreme learning machines," in *2017 Asilomar Conference on Signals, Systems, and Computers*, IEEE, Oct. 2017.

[96] E. Cambria, G.-B. Huang, L. L. C. Kasun, H. Zhou, C. M. Vong, J. Lin, J. Yin, Z. Cai, Q. Liu, K. Li, *et al.*, "Extreme learning machines [trends & controversies]," *IEEE Intelligent Systems*, vol. 28, no. 6, pp. 30–59, 2013.

[97] P. C. Loizou, *Speech enhancement : theory and practice*, ser. Signal processing and communications. Boca Raton: CRC Press, 2007, 608 p. ISBN: 9780849350320 (alk. paper) 0849350328 (alk. paper).

[98] E. Rothauser, W. Chapman, N Guttman, K. Nordby, H. Silbiger, G. Urbanek, and M Weinstock, "IEEE recommended practice for speech quality measurements," *IEEE Trans. Audio Electroacoust*, vol. 17, no. 3, pp. 225–246, 1969.

[99] G. Hu, *A corpus of nonspeech sounds*, [Online; accessed 13-September-2016].

[100] L. R. Rabiner and R. W. Schafer, *Theory and applications of digital speech processing*. Pearson, 2011.

[101] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, IEEE, vol. 2, 2001, pp. 749–752.

[102] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, and I. Vlahavas, "Multi-target regression via input space expansion: Treating targets as inputs," *Machine Learning*, vol. 104, no. 1, pp. 55–98, 2016.

[103] W. Krzanowski, *Principles of multivariate analysis*. OUP Oxford, 2000, vol. 23.

[104] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the royal statistical society. Series B (Methodological)*, pp. 111–147, 1974.

[105] D. Dheeru and E. Karra Taniskidou, *UCI machine learning repository*, 2017.

[106] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "Mulan: A Java library for multi-label learning," *Journal of Machine Learning Research*, vol. 12, pp. 2411–2414, 2011.

[107] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks.," in *Aistats*, vol. 15, 2011, p. 275.

[108] M. D. Zeiler, M Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, *et al.*, "On rectified linear units for speech processing," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, IEEE, 2013, pp. 3517–3521.

[109] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[110] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015.

[111] D. Wang, P. Wang, and Y. Ji, "An oscillation bound of the generalization performance of extreme learning machine and corresponding analysis," *Neurocomputing*, vol. 151, pp. 883–890, 2015.

[112] P. A. Henriquez and G. A. Ruz, "An empirical study of the hidden matrix rank for neural networks with random weights," in *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*, IEEE, 2017, pp. 883–888.

[113] B. O. Odelowo and D. V. Anderson, "A noise prediction and time-domain subtraction approach to deep neural network based speech enhancement," in *2017 16th*

IEEE International Conference on Machine Learning and Applications (ICMLA), Dec. 2017, pp. 372–377.

[114] D Wang and J. Lim, "The unimportance of phase in speech enhancement," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, no. 4, pp. 679–681, 1982.

[115] B. J. Shannon and K. K. Paliwal, "Role of phase estimation in speech enhancement," in *Ninth International Conference on Spoken Language Processing*, 2006.

[116] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *speech communication*, vol. 53, no. 4, pp. 465–494, 2011.

[117] A. Sugiyama and R. Miyahara, "Phase randomization-a new paradigm for single-channel signal enhancement," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, IEEE, 2013, pp. 7487–7491.

[118] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, IEEE, 2010, pp. 4214–4217.

[119] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Transactions on audio, speech, and language processing*, vol. 16, no. 1, pp. 229–238, 2008.

[120] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[121] B. O. Odelowo and D. V. Anderson, "A study of training targets for deep neural network-based speech enhancement using noise prediction," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, Apr. 2018.

[122] ——, "A mask-based post processing approach for improving the quality and intelligibility of deep neural network enhanced speech," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2017, pp. 1134–1138.

[123] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "Dynamic noise aware training for speech enhancement based on deep neural networks," in *INTERSPEECH*, 2014, pp. 2670–2674.

[124] C. Plapous, C. Marro, and P. Scalart, "Speech enhancement using harmonic regeneration," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, IEEE, vol. 1, 2005, pp. I–157.

[125] H. Ding, Y. Soon, S. N. Koh, and C. K. Yeo, "A post-processing technique for regeneration of over-attenuated speech," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, IEEE, 2009, pp. 3889–3892.

[126] E. Zavarehei, S. Vaseghi, and Q. Yan, "Noisy speech enhancement using harmonic-noise model and codebook-based post-processing," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1194–1203, 2007.

[127] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "Global variance equalization for improving deep neural network based speech enhancement," in *Signal and Information Processing (ChinaSIP), 2014 IEEE China Summit & International Conference on*, IEEE, 2014, pp. 71–75.

[128] Y. Xu, J. Du, Z. Huang, L.-R. Dai, and C.-H. Lee, "Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement," *arXiv preprint arXiv:1703.07172*, 2017.

[129] D. Wang and G. J. Brown, *Computational auditory scene analysis: Principles, algorithms, and applications*. Wiley-IEEE Press, 2006.

[130] M. Kutner, C. Nachtsheim, J. Neter, and W. Li, *Applied linear statistical models*. McGraw Hill, 2004.

[131] J. W. Tukey, "Comparing individual means in the analysis of variance," *Biometrics*, pp. 99–114, 1949.

[132] L. Xu, C.-S. Choy, and Y.-W. Li, "Deep sparse rectifier neural networks for speech denoising," in *Acoustic Signal Enhancement (IWAENC), 2016 IEEE International Workshop on*, IEEE, 2016, pp. 1–5.

[133] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vong, "Representational learning with ELMs for big data," 2013.

[134] W. Yu, F. Zhuang, Q. He, and Z. Shi, "Learning deep representations via extreme learning machines," *Neurocomputing*, vol. 149, pp. 308–315, 2015.

[135] M.-B. Li, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "Fully complex extreme learning machine," *Neurocomputing*, vol. 68, pp. 306–314, 2005.

[136] M. D. McDonnell, M. D. Tissera, T. Vladusich, A. Van Schaik, and J. Tapson, "Fast, simple and accurate handwritten digit classification by training shallow neural network classifiers with the extreme learning machinealgorithm," *PloS one*, vol. 10, no. 8, e0134254, 2015.

[137] A. G. Pacheco, R. A. Krohling, and C. A. da Silva, "Restricted Boltzmann machine to determine the input weights for extreme learning machines," *Expert Systems with Applications*, vol. 96, pp. 77–85, 2018.

[138] H. Erdogan, J. R. Hershey, S. Watanabe, and J. L. Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 708–712, ISBN: 1520-6149.

[139] A. Kumar and D. Florencio, "Speech enhancement in multiple-noise conditions using deep neural networks," *arXiv preprint arXiv:1605.02427*, 2016.

[140] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," 1999.

[141] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[142] S. Pascual, A. Bonafonte, and J. Serra, "SEGAN: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[143] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Prentice-Hall, 1999.

[144] J. A. Moorer, "A note on the implementation of audio processing by short-term Fourier transform," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct. 2017, pp. 156–159.

[145] J. I. Marín-Hurtado and D. V. Anderson, "FFT-based block processing in speech enhancement: Potential artifacts and solutions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2527–2537, 2011.

[146] S. Džeroski, D. Demšar, and J. Grbović, "Predicting chemical parameters of river water quality from bioindicator data," *Applied Intelligence*, vol. 13, no. 1, pp. 7–17, 2000.

[147] W. Groves and M. Gini, "On optimizing airline ticket purchase timing," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 7, no. 1, p. 3, 2015.

[148] A. Karalič and I. Bratko, "First order regression," *Machine Learning*, vol. 26, no. 2-3, pp. 147–176, 1997.

[149] A. Tsanas and A. Xifara, "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools," *Energy and Buildings*, vol. 49, pp. 560–567, 2012.

[150] W. Groves and M. Gini, "Improving prediction in TAC SCM by integrating multivariate and temporal aspects via PLS regression," in *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, Springer, 2013, pp. 28–43.

[151] I.-C. Yeh, "Modeling slump flow of concrete using second-order regressions and artificial neural networks," *Cement and Concrete Composites*, vol. 29, no. 6, pp. 474–480, 2007.

VITA

Babafemi Odelowo was born in Washington, D.C. He received a B.Sc in Electronic and Electrical Engineering from the Obafemi Awolowo University, Ile-Ife, Osun State, Nigeria, before returning to the United States for further education. He earned a M.S degree from the Department of Electrical and Computer Engineering at the Georgia Institute of Technology (Georgia Tech) before proceeding to industry where he worked for the Naval Undersea Warfare Center, Newport, Rhode Island and Northrop Grumman Integrated Systems, Bethpage, New York. He returned to Georgia Tech and earned a M.S in Operations Research prior to joining the Efficient Signal Processing Laboratory where he performed research on the use of machine learning methods in signal processing. His research interests include statistical signal processing, machine learning, and speech/audio signal processing.